

# Apple



# Assembly Line

\$1.80

Volume 8 -- Issue 1

October, 1987

Key-Edit Utility Program for the IIgs. . . . .	2
INDEX to Volume 7 (October 1986 thru September 1987) . . . . .	7
How to Clear the ProDOS-8 Back-Up Bit. . . . .	26
Sean Nolan's Proposed ProDOS-8 SETUP File Standard . . . . .	29

### Thanksgiving

I know the date above says October, but it is now the day before Thanksgiving on my calendar. Since Bill Morgan moved to a new job in Austin, Texas, it has been quite a challenge to put the newsletter together! I am thankful that you are patient, and that God is faithful. And He reigns. Even in this day of wild stock markets, economic uncertainty, natural disasters, and wars we can see that He is, and that "He has made the earth by his power, He has established the world by His wisdom, and has stretched out the heaven by His understanding."

### Using Laumer Research Full Screen Editor on IIgs

The ProDOS version of the Full Screen Editor is compatible with the IIgs and with Version 2.0 of the S-C Macro Assembler. The DOS version, on the other hand, requires a different setup file. This is now provided as the file "ED IIGS 2.0", a text file which you EXEC to load and install the FSE. Since there are a lot of you who may need this file so I will describe its contents here:

```
MNTR
BLOAD EDITOR,A$5000
BLOAD B.DRIVER.AP2E,A$5DB5
0=L
0/D000<0/5000.5FFFM
1=L
0/D074:4C FA F7
0/F7F4:AD 80 C0 4C 03 D0 AD 88 C0 4C 00 D0
Q
```

You can create this file by typing the above lines with line numbers in the S-C Macro Assembler, and then save them on a text file using the command "TEXT ED IIGS 2.0". You may of course move the files ED IIGS 2.0, EDITOR, and B.DRIVER.AP2E to your assembler disks, and create a composite load file so that the FSE installation is automatic.

## Key-Edit Utility Program for the IIgs.....Bob Boughner

Key-Edit is a IIgs utility program that permits the rapid recall and easy editing of previously entered commands or program source lines. Key-Edit works in either 40- or 80-column mode, and will work inside a window smaller than the full screen if you set the proper values in the monitor window definition bytes. The program is patterned after a similar program I use at work on an IBM AT, which explains some of the rather strange keyboard choices.

The current version, shown here, of Key-Edit works only under DOS. I hope to modify it for at least ProDOS-8 in the near future. Because of the keyboard choices I made, it is currently tied to the IIgs. Toward the end of this article I will describe the changes needed to make it work in a //e or //c.

Key-Edit requires four pages (1024 bytes) for itself and the buffer of previously-typed commands. I obtained these four pages by moving the DOS buffers down (an old trick). When you BRUN KEY.EDIT it loads at \$5000; the code at \$5000 then moves the buffers down four pages, copies Key-Edit into the hole starting at \$9900, and clears a command buffer at \$9C00-\$9CFF. I also patch in a RESET routine so that RESET will re-install Key-Edit. The vector which was at \$3F2,3F3 is copied into my patch, so that RESET will still function normally. Of course if you later patch a new RESET vector, my patch will be left out in the cold.

Key-Edit works by intercepting the normal keyboard input stream. A table (lines 2660-2870) defines fifteen special combinations of normal keys with the Open-Apple, Option (Solid-Apple), Control, and Shift keys. If any of these special keystrokes is detected, Key-Edit performs an editing operation on the input buffer and the screen. Other keystrokes are passed to the program which asked for input. Key-Edit is compatible with the S-C Macro Assembler and Applesoft, and may work with other DOS-based applications.

When you finally type a RETURN, the current line is copied into my one-page buffer. A pointer keeps track of the end of the last line copied in my buffer, so each command is copied onto the end of the previous command. I separate the commands in my buffer with \$00 bytes rather than RETURN (\$8D) codes, because it is easier to test for them in the various loops. When you type the "up arrow" key, the most recently entered command line is retrieved, placed into the buffer at \$200, and displayed on the screen. Another "up arrow" will move back in time to the command line before that one, and so on. Since the buffer is only 256 bytes long, there is a limit to how many commands are remembered. The buffer is circular, so after you have looked at the oldest one another "up arrow" will show you the newest one. The "down arrow" key moves you around the buffer in the opposite direction. When you find the command you want, simply hit RETURN to execute it. Or, if it is only approximately what you want, edit it and then hit RETURN.

S-C Macro Assembler Version 2.0.....DOS	\$100,	ProDOS	\$100,	both for \$120
Version 2.0 DOS Upgrade Kit for 1.0/1.1/1.2 owners.....				\$20
ProDOS Upgrade Kit for Version 2.0 DOS owners.....				\$30
Source Code of S-C Macro 2.0 (DOS or ProDOS).....each, additional \$100				
S-C Disassembler (ProDOS only)....without source code \$30, with source \$50				
RAK-Ware DISASM (DOS only).....without source code \$30, with source \$50				
PROVIEW (ProDOS only).....				\$20
Full Screen Editor For S-C Macro (with complete source code).....				\$49
S-C Cross Reference Utility.....without source code \$20, with source \$50				
S-C Word Processor (with complete source code).....				\$50
DP18 and DPFP, including complete source and object code.....				\$50
ES-CAPE (Extended S-C Applesoft Program Editor).....				
Including Version 2.0 With Source Code.....				\$50
ES-CAPE Version 2.0 and Source Code Update (for Registered Owners).....				\$30
Bag of Tricks 2 (Quality Software).....{(\$49.95)} \$45				
MacASM -- Macro Assembler for Macintosh (Mainstay).....{(\$150.00)} \$50				
S-C Documentor (complete commented source code of Applesoft ROMs).....\$50				
Cross Assemblers for owners of S-C Macro Assembler.....\$32.50 to \$50 each				
(Available: 6800/1/2, 6301, 6805, 6809, 68000, Z-80, Z-8, 8048,				
8051, 8085, 1802/4/5 PDP-11, GI1650/70, others)				
Beagle Bros. Applesoft Compiler (ProBOS only).....				\$74.95)
Copy II Plus (Central Point Software).....				{(\$39.95)} \$30
SoftSwitch (Roger Wagner Publishing).....				\$59.95)
AAL Quarterly Disks.....			each \$15, or any four for \$45	
Each disk contains the source code from three issues of AAL:				
Jan-Mar, Apr-Jun, Jul-Sep, and Oct-Dec.				
(All source code is formatted for S-C Macro Assembler. Other assemblers				
require some effort to convert file type and edit directives.)				
Diskettes (with hub rings, sleeves, labels).....		box of 10 for \$6		
Vinyl disk pages, 6"x8.5", hold two disks each).....		10 for \$6		
Diskette Mailing Protectors (hold 1 or 2 disks).....		40 cents each		
(Cardboard folders designed to fit 6"X9" Envelopes.)		or \$25 per 100		
Envelopes for Diskette Mailers.....		6 cents each		
Sider 20 Meg Hard Disk, includes controller & software.....		(\$695)	\$550	
Minuteman 250 Uninterruptible Power Supply.....		{(\$359)}	\$320	
65802 Microprocessor (Western Design Center).....			\$25	
quikLoader EPROM System (SCRG).....			{(\$179)} \$170	
PROmGamer (SCRG).....			{(\$149.50)} \$140	
"Exploring the Apple IIgs", Gary B. Little.....		\$22.95)	\$21	
"Apple IIgs Technical Reference", Fischer.....		\$19.95)	\$19	
"65816/65802 Assembly Language Programming", Fischer.....		\$21.95	\$20	
"Programming the 65816", Eyes & Lichty.....		\$22.95	\$21	
"App. //e Reference Manual", Apple Computer.....		\$24.95	\$23	
"App. //c Reference Manual", Apple Computer.....		\$24.95	\$23	
"Prot S-8 Technical Reference Manual", Apple Computer.....		\$29.95	\$27	
"Prot S-16 Technical Reference Manual", Apple Computer.....		\$29.95	\$27	
"App. IIgs Firmware Reference", Apple Computer.....		\$24.95	\$23	
"App. IIgs Hardware Reference", Apple Computer.....		\$24.95	\$23	
"Prot S Inside and Out", Doms & Weishaar.....		\$16.95	\$16	
"DOS 1.1k Scrapbook", Weishaar & Kersey.....		\$14.95	\$14	
"Beneath Apple ProDOS", Worth & Lechner.....		\$19.95	\$18	
"Beneath Apple DOS", Worth & Lechner.....		\$19.95	\$18	
"Inside the Apple //e", Little.....		\$19.95	\$18	
"Inside the Apple //e", Little.....		\$19.95	\$18	
"Understanding the Apple //e", Sather.....		\$24.95	\$23	
"Understanding the Apple II", Sather.....		\$22.95	\$21	
"App. II/IIe Troubleshooting & Repair Guide", Brenner.....		\$19.95	\$18	
"Assembly Language for Applesoft Programmers", Finley & Myers.....		\$18.95	\$18	
"Now That You Know Apple Assembly Language...", Gilder.....		\$19.95	\$18	
"Enhancing Your Apple II, vol. 1", Lancaster.....		\$15.95	\$15	
"Enhancing Your Apple II, vol. 2", Lancaster.....		\$17.95	\$17	
"Assembly Cookbook for the Apple II/IIe", Lancaster.....		\$21.95	\$20	
"Microcomputer Graphics", Myers.....		\$14.95	\$14	
"Assembly Lines -- the Book", Wagner.....		\$19.95	\$12	

\* On these items add \$2.00 for the first item and \$.75 for each additional item for US shipping.

+ Inquire for shipping cost.

Foreign customers inquire for postage needed.

Texas residents please add 8% sales tax to all orders.

\*\*\* S-C SOFTWARE, P. O. BOX 280300, Dallas, TX 75228 \*\*\*

When you start up Key-Edit, the command buffer is cleared. Each time you enter a new command, it is copied on the end of the previous command. Once the end of the buffer is reached, the buffer wraps around to the beginning, and the command there is over-written. The over-written command is obviously the oldest command in the buffer. Then enough \$00 bytes are stored to clear out any partial command line. This allows the up- and down-arrows to scan smoothly forward or backward around the buffer.

The editing commands in Key-Edit allow you to scan back and forth along the input line, delete characters, insert characters, and submit the line without needing to scan to the end-of-line first. Regardless of where the cursor is on the line, you can type the RETURN key and the whole line will be submitted. Scanning back and forth is done with either the TAB key or the left and right arrow keys, with or without the open-apple key:

left arrow	move back one character
right arrow	move ahead one character
OA-left	move to beginning of line
OA-right	move to end of line
TAB	move ahead to next "tab" stop
OA-TAB	move back to next "tab" stop

These keys will not move backward beyond the beginning of the line, nor forward past the end of the line. A "tab" stop is defined as the next occurrence of any of five characters: space, comma, period, semi-colon, or colon. The cursor will move until it encounters one of those five characters or the end or beginning of the line. The five "tab" characters are contained in a table at line 5710, so you modify the list as you wish.

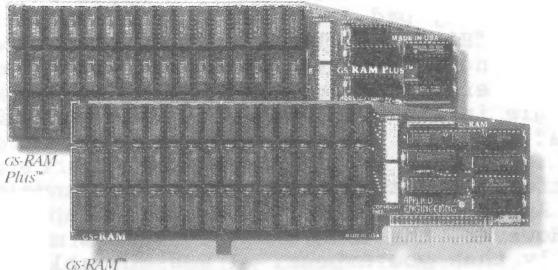
There are six different commands available to delete characters from the current line. The simplest is simply the DELETE key, which deletes the key to the left of the cursor. If the cursor is at the beginning of the line, the first character is deleted. Pressing the Open Apple with the DELETE key deletes the entire line. Pressing the Control-key with DELETE deletes from the cursor to the end of line. Control-Shift-DELETE scrubs everything from the left of the cursor to the beginning of the line. Option (or Solid-Apple) with DELETE causes my internal command buffer to be cleared. Finally, hitting the decimal point key on the numeric key pad with the Open-Apple key deletes the key under the cursor. Here they are in tabular format:

DELETE	Delete char before cursor
OA-DEL	Delete entire line
C-DEL	Delete to end of line
CS-DEL	Delete to beginning
OP-DEL	Clear command buffer
OA-PAD-".."	Delete char under cursor

That last one is there because of the program I use at work on my IBM AT. It has a DELETE CHAR key on the key pad decimal

# For Those Who Want the Most From Those Who Make the Best. GS-RAM™

Now expand the IIgs' RAM and ROM with up to 8 MEG of "Instant On" memory with the all new GS-RAM!



gs-RAM has an all new design. A design that delivers higher performance including increased speed, greater expandability, and improved software.

## More Sophisticated, Yet Easier to Use

Gs-RAM works with all IIgs software. In fact any program that runs on Apple's smaller memory card runs on the gs-RAM. But with gs-RAM you can have more memory, improved performance, and almost unlimited expansion capabilities. We've designed the new gs-RAM to be easier to use too—you don't have to adjust the size of your RAM disk every time you use a DMA device. No other RAM card with more than 4 banks of memory installed can make the same claim.

## More than just Hardware

Each gs-RAM and gs-RAM Plus includes the most powerful set of IIgs software enhancements available anywhere. In fact, our nearest competitor offers only a fraction of the invaluable programs that we include with each gs-RAM card. This software includes the most powerful disk-caching program available, the gs-RAM Cache. The Cache will make most of your applications run up to *7 times faster*. Also included is a diagnostic utility that lets you test your gs-RAM by graphically showing the location of any bad or improperly installed RAM chips. And for AppleWorks users, we give you our exclusive Expander program that dramatically enhances both the capabilities and speed of AppleWorks.

## Making AppleWorks Even Better

Applied Engineering's Expander program eliminates AppleWorks internal memory limits allowing it to recognize up to 8 megabytes of Desktop workspace. You can increase the limit from only 7,250 lines to 22,600 lines in the Word Processor and from 6,350 records to 22,600 records in the Database. The Expander allows all of AppleWorks, including print functions, to automatically load into RAM. The clipboard size will increase from 255 to 2,042 lines maximum. gs-RAM will automatically segment larger files so you can save them onto multiple floppies. And

gs-RAM provides a built-in print buffer that allows you to continue working in AppleWorks while your printer is still processing text. You can even load Pinpoint or MacroWorks and your favorite spelling checker into RAM for instant response.

## Grow by Kilobytes or Megabytes

We offer gs-RAM in two configurations so you can increase your memory 256K at a time (gs-RAM) or a megabyte at a time (gs-RAM Plus). Both are IIgs compatible and both come with our powerful enhancement software. gs-RAM can hold up to 1.5 MEG of 256K chips and gs-RAM Plus can hold up to 6 MEG using 1 MEG chips. And since both use standard RAM chips (not high-priced SIMMs), you'll find expanding your gs-RAM or gs-RAM Plus easy, convenient, and very economical. For further expansion you can plug a 2 MEG "piggyback" card into the gs-RAM's expansion port for up to 3.5 MEG of total capacity. Or up to a whopping 8 MEG on gs-RAM Plus. If a gs-RAM owner outgrows 3.5 MEG, he can easily upgrade to gs-RAM Plus for a nominal charge.

## Permanent Storage for an "Instant On" Apple

With our RamKeeper™ back-up option, your gs-RAM or gs-RAM Plus will retain both programs and data while your IIgs is turned off! Now when you turn your IIgs back on, your favorite software is on your screen in under 4 seconds! With RamKeeper you can divide your IIgs memory into part "electronic hard disk" and part extended RAM. Even change the memory boundaries at any time—and in any way—you want. Because



Steve Jobs, the creator of Apple Computer

"In quality, performance, compatibility, expandability and support, Applied Engineering's gs-RAM and gs-RAM Plus are number one."

Applied Engineering has the most experience in the industry with battery-backed memory for the Apple, you are assured of the most reliable memory back-up system available. And in the world of battery-backed memory, Reliability is everything. That's why Applied Engineering uses state-of-the-art "GEL CELLS" instead of Ni-Cads batteries (if Ni-Cads aren't discharged periodically, they lose much of their capacity). RamKeeper has about 6 hours of "total power failure" back-up time. That's 6 times the amount of other systems. But with power from your wall outlet, RamKeeper will back-up gs-RAM, gs-RAM Plus, or most other IIgs memory cards indefinitely. Should you ever have a "total power failure," RamKeeper switches to its 6-hour battery. When power returns, RamKeeper will automatically recharge the battery to full power. RamKeeper incorporates a dual-rate charger, status LED's, and advanced power reducing circuitry. RamKeeper comes complete with battery, software, and documentation.

## GS-RAM's Got it ALL!

- 5-year hassle-free warranty
- 6 RAM banks (most cards have 4)
- Memory expansion port
- ROM expansion port
- Ultra-fast disk caching on ProDOS 8 AND ProDOS 16.
- Expands AppleWorks internal limits
- Includes hi-res self test
- No soldered-in RAM chips
- Expandable to 8 MEG
- No configuration blocks to set
- RamKeeper back-up option allows permanent storage of programs & data
- 15-day money-back guarantee
- Proudly made in the USA

gs-RAM with 256K	\$169
gs-RAM with 512K	\$219
gs-RAM with 1 MEG	\$299
gs-RAM with 1.5 MEG	\$379
gs-RAM with 2-8 MEG	CALL
gs-RAM Plus with 1-8 MEG	CALL
RamKeeper option	\$179

## Order today!

See your dealer or call Applied Engineering today, 9 am to 11 pm, 7 days. Or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 7% sales tax. Add \$10.00 outside U.S.A.

**A** APPLIED ENGINEERING™  
The Apple enhancement experts.

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

Prices subject to change without notice.

point, and INSERT TOGGLE key on the key pad "0" key. If you do not have a numeric key pad, change the equate line 1010 to "HAVE.PAD .EQ 0"; then it will assemble Open-Apple Period for the DELETE CHAR function, and Open-Apple Comma for the INSERT TOGGLE function.

Speaking of INSERT.... If you hold down the Open-Apple and type the "0" on the numeric key pad (or comma on the regular keyboard if you select the "HAVE.PAD .EQ 0" option), the cursor will be changed to a flashing `~'. Then any normal characters you type will be inserted before the cursor. Typing the OA-PAD-0 again will toggle insert mode back off, and you will be back in the "replace" mode.

Key-Edit does not interfere with the other editing commands which are available in Applesoft and the S-C Macro Assembler, it supplements them. However, the ESCAPE key functions may work a little differently than you expect. I was not able to get them to work exactly as they should. Maybe you can do it.

Since Key-Edit uses the input hook, it can be disconnected by typing PR#3 (which engages both input and output hooks and turns on 40-columns), or any IN# command. Hitting RESET will re-connect it. Remember that the DOS buffers have been moved down to make space for Key-Edit. If you move them back up, be sure you also unhook the input vector change the RESET vector to no longer point into Key-Edit, as it will be G-O-N-E.

Lines 1210-1580 are the code which install Key-Edit. Lines 1210-1230 check to see if it is already installed, and if so just return without doing anything. Lines 1600-1870 are an alternate installation routine which I used during development. I deleted the .TF at line 1070, so that assembly was into RAM at \$5000. Then the first time I tested Key-Edit I typed \$5000G to install it. After subsequent patches and re-assembly, I typed MGOT to copy the new code into place. If you start making changes, you can use the same technique. In fact, you might want to change the "BEQ .6" at line 1230 to "BEQ T", to make it semi-automatic.

The command table (lines 2660-2870) uses a macro to simplify typing it in. The listing shows the macro expansion, so you can type them directly if you wish. The first byte of each four-byte entry is the ASCII value of the character. The second byte is the contents of the keyboard modifier register, which is \$C025 in the Apple IIgs. The eight bits of this byte tell you which modifier keys were pressed when the key was typed:

```
bit 7: Open-Apple
bit 6: Option (Solid Apple)
bit 5: (data updated without keypress)
bit 4: numeric key pad
bit 3: repeat active
bit 2: caps lock down
bit 1: control key down
bit 0: shift key down
```

(this article cont'd page 15)

## INDEX

### Apple Assembly Line, Volume 7 October, 1986 through September, 1987

**AAAA**

Address Booklet, Printing a Tiny.....	Bob S-C...	6/87/24-32
<b>Apple IIgs</b>		
A New BASIC for the Apple IIgs (GS BASIC).....	Bob S-C...	9/87/24
A New Pattern Search Monitor Command.....	Bob S-C...	5/87/2-11
Another ProDOS-8 Bug in the IIgs.....	Bob S-C...	7/87/19-22
Apple Finally Lifts the Lid on the IIgs.....	Bob S-C...	10/86/2-8
Beginning to Peek at the IIgs Monitor.....	Bob S-C...	12/86/21-24
Clearing Some Mist from Super Hi-Res.....	Bob S-C...	7/87/2-9
Demo of Two Simple IIgs Tools.....	Bob S-C...	12/86/27-29
Display the IIgs Monitor Command Table.....	Bob S-C...	12/86/21-24
Display the IIgs Tool Tables.....	Bob S-C...	2/87/21-23
Display the 65802 Registers (65816 too).....	Bob S-C...	2/87/25-29
IIgs Battery RAM and Clock/Calendar.....	Bob S-C...	11/86/24-32
IIgs I/O Soft Switches.....	Bob S-C...	1/87/2-7
IIgs ROM Checksum.....	Bob S-C...	12/86/9
IIgs Tool Set Version Numbers.....	Bob S-C...	5/87/29-32
Monitor Ctrl-Y Command in the IIgs.....	Bob S-C...	1/87/12-15
More Than Just Saving the Registers.....	Bob S-C...	6/87/12-18
New ROM-chip upgrade for the IIgs.....		8/87/1
New Features in Apple IIgs Monitor.....	Bob S-C...	1/87/23-27
Notes on the IIgs Monitor.....	Sandy Mossberg...	3/87/5-7
Problem with IIgs 3.5"-Drive Firmware.....	Bob S-C...	5/87/26-27
Reading the IIgs ROMs.....	Bob S-C...	4/87/13
Some IIgs Demos.....	Bob Urschel...	9/87/11-16
Warning to Smart Port Programmers.....	Tom Vier...	7/87/15
Who Worked on the IIgs?.....	Bob S-C...	12/86/9-10

**Applesoft**

Beagle Bros. Applesoft Compiler Review.....	Bill Morgan...	3/87/8-14
EXEC and INPUT Bug in BASIC.SYSTEM.....	Bob S-C...	7/87/23
Selective CLEAR for Applesoft Variables.....	Bruce E. Howell...	6/87/2-10

**AppleWorks**

Viewing Appleworks Word Processor Files.....	Bob S-C...	8/87/15-19
--	------------	------------

**Arithmetic**

ABC#DE=FG#HI Puzzle Solved.....	Bob S-C...	5/87/18-25
Signed and Unsigned Comparisons.....	Bob S-C...	7/87/25-32
Square Roots, Ultra-Fast Integer.....	Chas. Putney & Bob S-C...	11/86/2-12

**BBBB**

**Book Reviews**

Apple IIgs Reference Manuals.....		11/86/19
"Apple IIgs Firmware Reference", by Apple.....		7/87/15
"Apple IIgs Hardware Reference", by Apple.....		9/87/1
"Apple IIgs ProDOS-16 Reference", by Apple.....		7/87/18
"Apple IIgs Technical Reference", Michael Fischer.....		2/87/1
"Exploring the Apple IIgs", by Gary Little.....		9/87/1
Latest from Don Lancaster: "Ask the Guru".....		6/87/1
New IIgs Books from Apple.....		7/87/1
New Supplement to "Beneath Apple ProDOS Available (V1.2&1.3)....		4/87/30

**Book Reviews, cont'd**

"ProDOS Inside and Out", Weishaar and Doms.....	12/86/1
Return of "Beneath Apple ProDOS".....	6/87/1
Updated Supplement to "Beneath Apple ProDOS".....	10/86/1

**Bugs**

\$48 and "G" Strike Again!.....	Bob S-C... 3/87/29
A B g in FID with 400K Volumes.....	Bob S-C... 3/87/30-32
Another ProDOS-8 Bug in the IIgs.....	Bob S-C... 7/87/19-22
EXE and INPUT Bug in BASIC.SYSTEM.....	Bob S-C... 7/87/23
Lit le Bugs in ProDOS /RAM.....	Bob S-C... 1/87/22
More about Patching Apple's ProDOS Releases.....	Bob S-C... 5/87/14-15
Correction to ProDOS Patcher.....	Bob S-C... 6/87/18
New ProDOS Bug and Fix.....	Bob S-C... 11/86/13
Pro lem with IIgs 3.5"-Drive Firmware.....	Bob S-C... 5/87/26-27
Revision to DOS 3.3 for UniDisk 3.5.....	Bill Morgan... 4/87/13
Sor Bugs in Apple's ProDOS Version 1.3.....	Bob S-C... 3/87/2-4
War ing to Smart Port Programmers.....	Tom Vier... 7/87/15

**CCCC****Commented Assembly Listings of Apple's Code**

Commented Listing of ProDOS, \$DE00-DEF2.....	Bob S-C... 12/86/12-19
Commented Listing of ProDOS, \$DEF3-DPE4.....	Bob S-C... 3/87/15-28
Comparisons, Signed and Unsigned.....	Bob S-C... 7/87/25-32
Conversions	

Friendly Decimal Print Subroutine.....	Jan Eugenides... 4/87/22-26
--	-----------------------------

**Cross Assemblers**

New Version of 6805 Cross Assembler.....	Bob S-C... 10/86/1
--	--------------------

**DDDD****Disassemblers**

S-C ProDOS DisASM, an Intelligent DisAssembler.....	Bob S-C... 2/87/32
DOS 3.3	

**Enhancements and Patches**

Improved & Safer Text I/O Speedup for DOS.....	Louis Pitz... 1/87/8-11
Pre-Allocating DOS Files (for Hi-Speed SAVEing)....	Bill Morgan... 2/87/2-11
Problem with IIgs 3.5"-Drive Firmware.....	Bob S-C... 5/87/26-27
Revision to DOS 3.3 for UniDisk 3.5.....	Bill Morgan... 4/87/13

**EEEE****Enhancements and Patches to S-C Macro Assembler**

How to Use the USR Command (DOS 3.3 \$D000 Version)....	Bob S-C... 1/87/18-21
PAUSE Directive for S-C Macro Assembler 2.0.....	Bill Morgan... 11/86/20-23
Modify S-C Macro for Overlay Assembly.....	Bob S-C... 8/87/20-28
Patch .TF to Assemble System Files.....	Bob S-C... 8/87/11-13



### SPECIAL !!! EXPANDED RAM/ROM BOARD: \$39.00

Similar to our \$30 RAM/ROM dev board described below Except this board has two sockets to hold your choice of 2-2K RAM, 2-2K ROM or even 2-4K ROM for a total of 8K. Mix RAM and ROM too. Although Apple limits access to only 2K at a time, soft switches provide convenient socket selection. Hard switches control defaults.

### IMPROVED !!! ][ IN A MAC (ver 2.0): \$75.00

Now includes faster graphics, UniDisk support and more! Bi-directional data transfers are a snap! This Apple II emulator runs DOS 3.3/PRODOS (including 6502 machine language routines) on a 512K MAC or MACPLUS. All Apple II features are supported such as HI/LO-RES graphics, 40/80 column text, language card and joystick. Also included: clock, RAM disk, keyboard buffer, on-screen HELP, access to the desk accessories and support for 4 logical disk drives. Includes 2 MAC diskettes (with emulation, communications and utility software, plus DOS 3.3 and PRODOS system masters, including Applesoft and Integer BASIC) and 1 Apple II diskette.

### SCREEN.GEN: \$35.00

Develop HI-RES screens for the Apple II on a Macintosh. Use MACPAINT (or any other application) on the MAC to create your Apple II screen. Then use SCREEN.GEN to transfer directly from the MAC to an Apple II (with SuperSerial card) or IIC. Includes Apple II diskette with transfer software plus fully commented SOURCE code.

### MIDI-MAGIC for Apple //C: \$49.00

Compatible with any MIDI equipped music keyboard, synthesizer, organ or piano. Package includes a MIDI-out cable (plugs directly into modem port - no modifications required!) and 6-song demo diskette. Large selection of digitized QRS player-piano music available for 19.00 per diskette (write for catalog). MIDI-MAGIC compatible with Apple II family using Passport MIDI card (or our own input/output card w/drum sync for only \$99.00).

### FONT DOWNLOADER & EDITOR: \$39.00

Turn your printer into a custom typesetter. Downloaded characters remain active while printer is powered. Use with any Word Processor program capable of sending ESC and control codes to printer. Switch back and forth easily between standard and custom fonts. Special functions (like expanded, compressed etc.) supported. Includes HIRES screen editor to create custom fonts and special graphics symbols. For Apple II, II+, //e. Specify printer: Apple Imagewriter, Apple Dot Matrix, C.Itoh 8510A (Prowriter), Epson FX 80/85, or Okidata 92/192.

\* FONT LIBRARY DISKETTE #1: \$19.00 contains lots of user-contributed fonts for all printers supported by the Font Downloader & Editor. Specify printer with order.

### DISASM 2.2e \$30.00 (\$50.00 with SOURCE Code)

Use this intelligent disassembler to investigate the inner workings of Apple II machine language programs. DISASM converts machine code into meaningful, symbolic source compatible with S-C, LISA, ToolKit and other assemblers. Handles data tables, displaced object code & even provides label substitution. Address-based triple cross reference generator included. DISASM is an invaluable machine language learning aid to both novice & expert alike. Don Lancaster says DISASM is "absolutely essential" in his ASSEMBLY COOKBOOK.

### The 'PERFORMER' CARD: \$39.00 (\$59.00 with SOURCE Code)

Converts a 'dumb' parallel printer I/F card into a 'smart' one. Simple command menu. Features include perforation skip, auto page numbering with date & title, large HIRES graphics & text screen dumps. Specify printer: MX-80 with Graftrax-80, MX-100, MX-80/100 with Graftraxplus, NEC 8092A, C.Itoh 8510 (Prowriter), OkiData 82A/83A with Okigraph & OkiData 92/93.

### 'MIRROR' ROM: \$25.00 (\$45.00 with SOURCE Code)

Communications ROM plugs directly into Novation's Apple-Cat Modem card. Basic modes: Dumb Terminal, Remote Console & Programmable Modem. Features include: selectable pulse or tone dialing, true dialtone detection, audible ring detect, ring-back, printer buffer, 80 col card & shift key mod support.

### RAM/ROM DEVELOPMENT BOARD: \$30.00

Plugs into any Apple slot. Holds one user-supplied 2Kx8 memory chip (6116 type RAM for program development or 2716 EPROM to keep your favorite routines on-line). Maps into \$Cn00-\$CnFF and \$C800-\$CFFF.

### C-PRINT For The APPLE //C: \$69.00

Connect standard parallel printers to an Apple //c serial port. Separate P/S included. Just plug in and print!

---

Unless otherwise specified, all Apple II diskettes are standard (not copy protected!) 3.3 DOS.

Avoid a \$3.00 handling charge by enclosing full payment with order. VISA/MC and COD phone orders OK.

RAK-WARE 41 Ralph Road W. Orange NJ 07052 (201) 325-1885



GGGG

**Graphics**

Clearing Some Mist from Super Hi-Res.....	Bob S-C...	7/87/2-9
Convert Hi-Res Pictures to Double Hi-Res.....	David C. Johnson...	10/86/9-14
Corrections to HR to DHR Converter.....	David C. Johnson...	12/86/26
Drawing Circles the Fast Way.....	Bob S-C...	9/87/2-10
Some IIgs Demos.....	Bob Urschel...	9/87/11-16
Using Double Lo-Res Graphics.....	Bob S-C...	8/87/2-10
Woz Re-Codes Hi-ResAddress Calculations...S.	Wozniak, Bob S-C...	12/86/29-32

HHHH

**Hardware Reviews**

ByteStalker Software Analyzer Card.....	4/87/1
New Apple Accessories Coming Soon.....	8/87/1

IIII

Index to AAL Volumes 4, 5, and 6.....	Bob S-C...	10/86/insert
Input from Keyboard		
Simple LIne-Input Subroutine.....	Bob S-C...	4/87/19-20

MMMM

**Macros**

Some Handy Move Macros.....	Bob S-C...	4/87/28-29
Message Printing		
80-Column Screen Display Subroutine.....	Bob S-C...	4/87/2-12
Monitor Information and Enhancements		
\$#8 and "G" Strike Again!.....	Bob S-C...	3/87/29
A New Pattern Search Monitor Command.....	Bob S-C...	5/87/2-11
Beginning to Peek at the IIgs Monitor.....	Bob S-C...	12/86/21-24
Display the IIgs Monitor Command Table.....	Bob S-C...	12/86/21-24
Monitor Ctrl-Y Command in the IIgs.....	Bob S-C...	1/87/12-15
New Features in Apple IIgs Monitor.....	Bob S-C...	1/87/23-27
Notes on the IIgs Monitor.....	Sandy Mossberg...	3/87/5-7
MOVE Macros, Some Handy.....	Bob S-C...	4/87/28-29

NNNN

**New Product Announcements**

A New BASIC for the Apple IIgs (GS BASIC).....	Bob S-C...	9/87/24
A Note about Chessmaster 2000.....		4/87/32
Beagle Bros. Applesoft Compiler Review.....	Bill Morgan...	3/87/8-14
ByteStalker Software Analyzer Card.....		4/87/1
Chip-Level Repair Service, Jim Davis's Beehive Enterprises.....		1/87/1
Code on Disk from "Beneath Apple ProDOS".....	Pete Johnson...	1/87/32
Disk Subscriptions to AAL include Article Text.....		3/87/1
New Apple Accessories Coming Soon.....		8/87/1
New DOS and ProDOS Versions of 6805 Cross Assembler...Bob	S-C...	10/86/1
New ProDOS Version of S-C XREF.....	Bob S-C...	1/87/29-32
New RAK-WARE Utility "ScreenGen".....	Bob Kovaus...	10/86/28

## New Products, cont'd

New Supplement to "Beneath Apple ProDOS Available (V1.2&1.3)....	4/87/30
ProVIEW, by Doug McComsey.....	Bob S-C... 2/87/1,19
QuikLoader Firmware Enhancement Adds ProDOS Capability.....	1/87/1
S-C ProDOS DisASM, an Intelligent DisAssembler.....	Bob S-C... 2/87/32
Updated Supplement to "Beneath Apple ProDOS".....	10/86/1

## PPPP

### Patches and Modifications to Other Software

A Bug in FID with 400K Volumes.....	Bob S-C... 3/87/30-32
Little Bugs in ProDOS /RAM.....	Bob S-C... 1/87/22
New ProDOS Bug and Fix.....	Bob S-C... 11/86/13
More about Patching Apple's ProDOS Releases.....	Bob S-C... 5/87/14-15
Correction to ProDOS Patcher.....	Bob S-C... 6/87/18
Problem with IIgs 3.5"-Drive Firmware.....	Bob S-C... 5/87/26-27
Revision to DOS 3.3 for UniDisk 3.5.....	Bill Morgan... 4/87/13
Printing a Tiny Address Booklet.....	Bob S-C... 6/87/24-32

### ProDOS

#### Commented Listings

Commented Listing of ProDOS, \$DE00-DEF2.....	Bob S-C... 12/86/12-19
Commented Listing of ProDOS, \$DEF3-DFE4.....	Bob S-C... 3/87/15-28

#### Patches

Another ProDOS-8 Bug in the IIgs.....	Bob S-C... 7/87/19-22
EXEC and INPUT Bug in BASIC.SYSTEM.....	Bob S-C... 7/87/23
Little Bugs in ProDOS /RAM.....	Bob S-C... 1/87/22
More about Patching Apple's ProDOS Releases.....	Bob S-C... 5/87/14-15
Correction to ProDOS Patcher.....	Bob S-C... 6/87/18
New ProDOS Bug and Fix.....	Bob S-C... 11/86/13
Some Bugs in Apple's ProDOS Version 1.3.....	Bob S-C... 3/87/2-4
ProDOS-Based Intelligent Disassembler.....	Bob S-C... 2/87/32
ProDOS Version of S-C XREF.....	Bob S-C... 1/87/29-32
QUIT or BYE Code	
Another Selector Note.....	Bob S-C... 10/86/26
Another Update to Bob's Program Selector.....	Bob S-C... 12/86/20

#### Puzzles

A+ September Puzzle Solved.....	Bob S-C... 7/87/10-15
ABC*DE=FG*HI Puzzle Solved.....	Bob S-C... 5/87/18-25
Eight Queens.....	Phil Goetz... 9/87/19-21
Woz's Five-digit Math Puzzler.....	Steve Wozniak & Bob S-C... 10/86/21-24

## QQQQ

### QUIT Code in ProDOS

Another Selector Note.....	Bob S-C... 10/86/26
Another Update to Bob's Program Selector.....	Bob S-C... 12/86/20

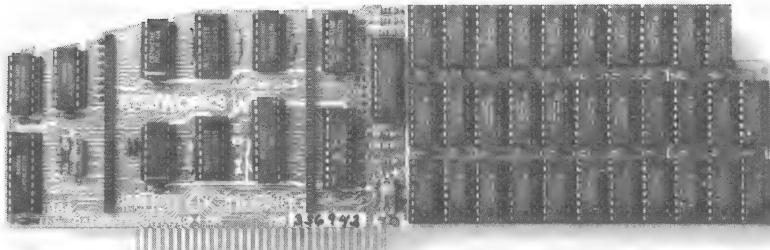
## RRRR

### RAM Disk

Little Bugs in ProDOS /RAM.....	Bob S-C... 1/87/22
Reviews, see "Book Reviews", "Hardware Reviews", "Software Reviews"	

# RamWorks® III

*Patented Performance from the Recognized Leader*



*The Best Selling, Most Compatible, Most Recommended, Most Expandable Card Available.*

#### **The AppleWorks Amplifier.**

While RamWorks III is recognized by all memory intensive programs, NO other expansion card comes close to offering the multitude of enhancements to AppleWorks that RamWorks III does. Naturally, you'd expect RamWorks III to expand the available desktop, after all Applied Engineering was a year ahead of everyone else *including Apple* in offering more than 55K and we still provide the largest AppleWorks desktops available. But a larger desktop is just part of the story. Look at all the AppleWorks enhancements that even Apple's own card does not provide and *only* RamWorks III does. With a 265K or larger RamWorks III, *all* of AppleWorks (including printer routines) will automatically load itself into RAM dramatically increasing speed by eliminating the time required to access the program disk drive. Switch from word processing to spreadsheet to database at the speed of light with no wear on disk drives.

RamWorks eliminates AppleWorks' internal memory limits, increasing the maximum number of records available from 1,350 to over 22,000. *Only* RamWorks increases the number of lines permitted in the word processing mode from 2,250 to over 22,000. *Only* RamWorks expands the maximum clipboard limit from 250 to 2,042 lines. And *only* RamWorks offers a built-in printer buffer, so you won't have to wait for your

printer to stop before returning to AppleWorks. And auto segments large files so they can be saved on two or more disks. You can even have Pinpoint or MacroWorks and your favorite spelling checker in RAM for instant response.

RamWorks, nothing comes close to enhancing AppleWorks so much.

#### **The Most Friendly, Most Compatible Card Available.**

Using RamWorks III couldn't be easier because it's compatible with more off-the-shelf software than any other RAM card. Popular programs like AppleWorks, Pinpoint, Catalyst, MouseDesk, HowardSoft, FlashCalc, Pro-FILER, Managing Your Money, SuperCalc 3a and VIP Professional to name a few (and most hardware add on's like ProFile and Sider hard disks). RamWorks is even compatible with software written for Apple cards. But unlike other cards, RamWorks plugs into the IIe auxiliary slot providing our super sharp 80 column text (U.S. Patent #4601018) in a completely integrated system while leaving expansion slots 1 through 7 available for other peripheral cards.

RamWorks III is compatible with all Apple IIe's, enhanced, unenhanced, American or European versions.

#### **Highest Memory Expansion.**

Applied Engineering has always offered the largest memory for the IIe and

RamWorks III continues that tradition by expanding to 1 full MEG on the main card using standard RAMs, more than most will ever need (1 MEG is about 500 pages of text)...but if you do ever need more than 1 MEG, RamWorks III has the widest selection of expander cards available. Additional 512K, 1 MEG, or 2 MEG cards just snap directly onto RamWorks III by plugging into the industry's only low profile (no slot 1 interference) fully decoded memory expansion connector.

#### **It Even Corrects Mistakes.**

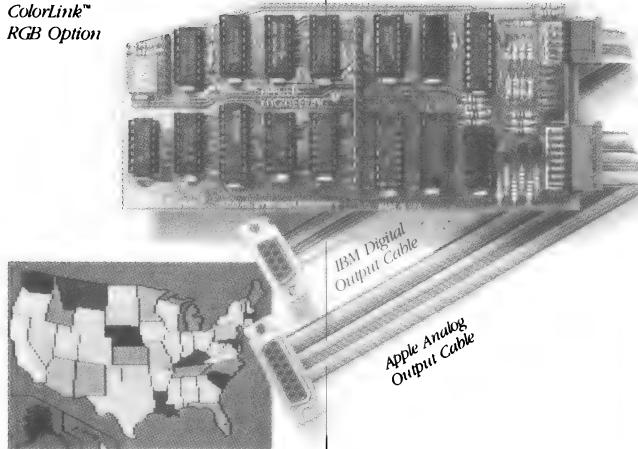
If you've got some other RAM card that's not being recognized by your programs, and you want RamWorks III, you're in luck. Because all you have to do is plug the memory chips from your current card into the expansion sockets on RamWorks to recapture most of your investment!

#### **The Ultimate in RGB Color.**

RGB color is an option on RamWorks and with good reason. Some others combine RGB color output with their memory cards, but that's unfair for those who don't need RGB *and* for those that do. Because if you don't need RGB Applied Engineering doesn't make you buy it, but if you want RGB output you're in for a nice surprise because the RamWorks RGB option offers better color graphics plus a more readable 80 column text (that blows away any

composite color monitor). For only \$129 it can be added to RamWorks giving you a razor sharp, vivid brilliance that most claim is the best they have ever seen. You'll also appreciate the multiple text colors (others only have green) that come standard. But the RamWorks RGB option is more than just the ultimate in color output because unlike others, it's fully compatible with all the Apple standards for RGB output control, making it more compatible with off-the-shelf software. With its FCC certified design, you can use almost any RGB monitor because only the new RamWorks RGB option provides both the new Apple standard analog and the IBM standard digital RGB outputs (cables included). The RGB option plugs into the back of RamWorks with no slot 1 interference and remember you can order the RGB option with your RamWorks or add it on at a later date.

#### *ColorLink™ RGB Option*



#### **True 65C816 16 Bit Power.**

RamWorks III has a built-in 65C816 CPU port for direct connection to our optional 65C816 card. The only one capable of linearly addressing more than 1 MEG of memory for power applications like running the Lotus 1-2-3™ compatible program, VIP Professional. Our 65C816 card does not use another slot but replaces the 65C02 while maintaining full 8 bit compatibility.

#### **Endorsed by the Experts.**

A+ magazine said "Applied Engineering's RamWorks is a boon to those who must use large files with AppleWorks...I like the product so much that I am buying one for my own system." in Cider

magazine said "RamWorks is the most powerful auxiliary slot memory card available for your IIe, and I rate it four stars...For my money, Applied Engineering's RamWorks is king of the hill."



*Steve Jobs, the creator  
of Apple Computer*

*"I wanted a  
memory card for  
my Apple that was  
fast, easy to use,  
and very compat-  
ible; so I bought  
RamWorks."*

Apple experts everywhere are impressed by RamWorks' expandability, versatility, ease of use, and the sheer power and speed that it adds to any IIe. With a RamWorks in your Apple, you'll make IBM PC's and AT's look like slowpokes.

- Can use 64K or 256K RAMs
- Powerful linear addressing 16 bit coprocessor port
- Automatic AppleWorks expansion up to 3017K desktop
- Accelerates AppleWorks
- Built-in AppleWorks printer buffer
- 100% compatible with all standard IIe software
- RamDrive™ and ProDrive™ the ultimate emulation software included free
- Memory is easily partitioned allowing many programs in memory at once
- Compatible RGB option featuring ultra high resolution color graphics and multiple text colors with cables for both the new Apple and standard IBM type monitors
- Self diagnostics software included
- Lowest power consumption (U.S. Patent #4601018)
- Takes only one slot (auxiliary) even when fully expanded
- Socketed and user upgradeable
- Software industry standard
- Advanced Computer Aided Design
- Used by Apple Computer, Steve Jobs and virtually all software companies
- Displays date and time on the AppleWorks screen with any PRO-DOS compatible clock
- Much, much more!

RamWorks III with 64K	\$179
RamWorks III with 256K	\$199
RamWorks III with 512K	\$249
RamWorks III with 1 MEG	\$329
RamWorks III with 1.5 MEG	\$489
RamWorks III with 3.0 MEG	\$1099
Ram Expander+ 1 MEG	\$499
Ram Expander+ 2 MEG	\$799
65C816 16 Bit Card	\$159
ColorLink RGB Option	\$129
Optional Software:	
Pinpoint with RAM Enhancement Software	\$79
VIP Professional	\$219

#### **Quality and Support of the Industry Leader.**

RamWorks III is from Applied Engineering, the largest, most well supported manufacturer of Apple peripherals and the inventor of large RAM cards for the Apple. With our 5 year no hassle warranty and outstanding technical support, you're assured of the most trouble free product you can buy.

#### **It's Got It All.**

- 15 day money back guarantee
- 5 year hassle free warranty
- Built-in super sharp 80 column display (U.S. Patent #4601018)
- Expandable to 1 MEG on main card
- Expansion connector for possible future expansion to 16 MEG

RamWorks III. The industry standard for memory expansion of the Apple IIe. ORDER YOUR RamWorks III TODAY. See your dealer or call (214) 241-6060, 9 a.m. to 11 p.m., 7 days or send check or money order to Applied Engineering.

MasterCard, Visa and C.O.D. welcome. Texas residents add 6 1/4% sales tax. Add \$10.00 if outside U.S.A.

**AE APPLIED ENGINEERING™**  
*The Apple enhancement experts*

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

## SSSS

### S-C Macro Assembler

Another ProDOS-8 Bug in the IIgs.....	Bob S-C...	7/87/19-22
Assembly Listings on ProDOS Text Files.....	John L. Hanna...	6/87/22-23
Modify S-C Macro for Overlay Assembly.....	Bob S-C...	8/87/20-28
Patch .TF to Assemble System Files.....	Bob S-C...	8/87/11-13
PAUSE Directive for S-C Macro Assembler 2.0.....	Bill Morgan...	11/86/20-23
Using the S-C Macro Assembler with the Echo Speech Synthesizer.....	Rick Hayner...	6/87/19-20

### Searching

A New Pattern Search Monitor Command.....	Bob S-C...	5/87/2-11
A Smarter Hexadecimal Memory Search.....	Bob S-C...	2/87/12-18

### Software Reviews

A New BASIC for the Apple IIgs (GS BASIC).....	Bob S-C...	9/87/24
A Note about Chessmaster 2000.....		4/87/32
Beagle Bros. Applesoft Compiler Review.....	Bill Morgan...	3/87/8-14
New DOS and ProDOS Versions of 6805 Cross Assembler...	Bob S-C...	10/86/1
New ProDOS Version of S-C XREF.....	Bob S-C...	1/87/29-32
New RAK-WARE Utility "ScreenGen".....	Bob Kovacs...	10/86/28
ProVIEW, by Doug McComsey.....	Bob S-C...	2/87/1,19
S-C ProDOS DisASM, an Intelligent DisAssembler.....	Bob S-C...	2/87/32

### Sounds

Nifty Sonic Signals.....	Bob S-C...	9/87/22-23
Some IIgs Demos.....	Bob Urschel...	9/87/11-16

### Source Code On Disk

Code on Disk from "Beneath Apple ProDOS".....	Pete Johnson...	1/87/32
Disk Subscriptions to AAL include Article Text.....		3/87/1

## TTTT

### Techniques

Eight Ways to Count Bits in a Byte.....	Bob S-C...	11/86/2-8
Friendly Decimal Print Subroutine.....	Jan Eugenides...	4/87/22-26
Signed and Unsigned Comparisons.....	Bob S-C...	7/87/25-32
Simple Line-Input Subroutine.....	Bob S-C...	4/87/19-20
Some Handy Move Macros.....	Bob S-C...	4/87/28-29

### Text Screen Subroutines

Moore Screen Tricks.....	Robert C. Moore...	10/86/15-16
Text Window Inverter.....	Bob S-C...	10/86/16-19
80-Column Screen Display Subroutine.....	Bob S-C...	4/87/2-12
Timing Apple Programs with Another Apple.....	Bob S-C...	11/86/15-19

## UUUU

### Unidisk 3.5

Problem with IIgs 3.5"-Drive Firmware.....	Bob S-C...	5/87/26-27
Revision to DOS 3.3 for UniDisk 3.5.....	Bill Morgan...	4/87/13

### Utility Programs

Printing a Tiny Address Booklet.....	Bob S-C...	6/87/24-32
Viewing AppleWorks Word Processing Files.....	Bob S-C...	8/87/15-19

WWWW

Windows

Text Window Inverter.....Bob S-C... 10/86/16-19  
Wozniak, Steve  
Woz's Five-digit Math Puzzler.....Steve Wozniak & Bob S-C... 10/86/21-24  
Woz Re-Codes Hi-ResAddress Calculations...S. Wozniak, Bob S-C... 12/86/29-32

65802/65816

Display the 65802 Registers (65816 too).....Bob S-C... 2/87/25-29  
More Than Just Saving the Registers.....Bob S-C... 6/87/12-18  
PER-fect Tool for 65816 Position-Independent Code.Sandy Mossberg.4/87/14-18  
Warning about 65816 JMP Indirect.....Bob S-C... 9/87/17-18

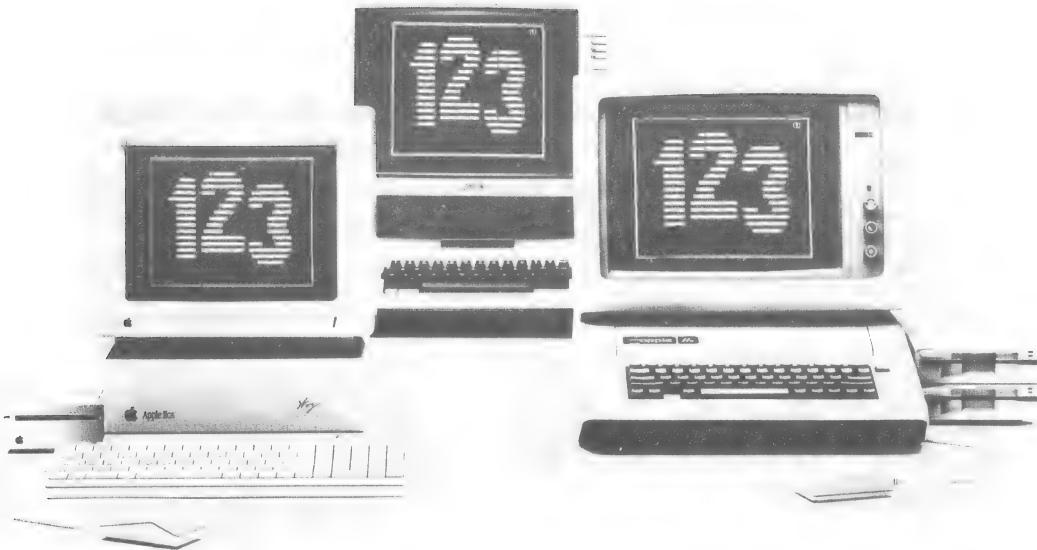
I only look at bits 7, 6, 4, 1, and 0. If you modify this program for a //e or //c, you will have to change all code which accesses \$C025 (called KEY.STATE in the program), because it doesn't exist in these older machines.

By modifying the keyboard command table and the insert-mode cursor, you could use Key-Edit on a //e or //c equipped with a 65802 processor. A little more work would eliminate the 65802-specific opcodes and let it run in a normal //c or enhanced //e with a 65C02 processor. In fact, I only used one opcode which is in the 65802 or 65816 and not in the 65C02, and I only used it one time: the TYX at line 5460. You could replace it with two lines: TYA and TAX. The insert-mode cursor would give a little more trouble, but you could just delete that code and leave the cursor alone. You have to modify the keyboard command table to use key combinations that are available and testable on the //e and //c. Then the PROCESS.CHAR subroutine would have to access the Open- and Solid-Apple information in a different way.

When I sent this program to AAL, it was nearly a full 3 pages plus the buffer and some patches inside DOS over the top of the INIT code. A few days later I thought of some changes and sent them on to AAL which shortened the code enough to eliminate the need for the patches inside DOS. However, the program as you see it here has been significantly revised by Bob Sander-Cederlof.

[Editor's note: Bob Boughner's code was good, but I just couldn't leave it alone. Blame any bugs you find on me! Hopefully you will be afflicted with the same impulse, and start modifying to your own taste.]

There is now room to add new features without expanding to more than a total of four pages. The end of code is currently at \$9BA7, and you can go on up to \$9BFF, so you have \$58 (88 decimal) bytes left to play in. If you go too far, lines 5980-6000 will catch you and signal the fact when you try to assemble.



# Now Apple speaks IBM. Three times faster than IBM.

**Introducing  
PC Transporter.<sup>TM</sup>**  
**The Apple<sup>®</sup> II expansion  
board that lets you  
run MS<sup>®</sup>-DOS programs.**

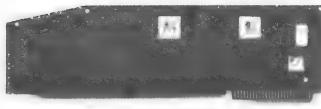
Now your Apple II can run over 10,000 programs you could never use before. Like Lotus<sup>®</sup> 1-2-3<sup>®</sup>, MultiMate<sup>®</sup>, dBASE III PLUS<sup>®</sup>, Even Flight Simulator<sup>®</sup>.

With PC Transporter, MS-DOS programs run on your Apple II like they do on IBM<sup>®</sup> PCs or compatibles. With one important difference. PC Transporter runs most of those programs *three times faster* than an IBM PC/XT<sup>®</sup>.

Plus, to speed through number-crunching tasks, you can use our optional 8087-2 math co-processor chip. It plugs into a socket on the PC Transporter.

**Less expensive than an  
IBM clone.**

Sure, a stripped-down IBM



clone costs about the same as the PC Transporter. But the peripherals it takes to get the clone up and running make the clone cost about three times what our American-made card costs.

You don't have to buy new hardware to use PC Transporter. **Works with the hardware you already own.**

With PC Transporter, MS-DOS programs see your Apple hardware as IBM hardware. You can use the same hardware you have now.

With IBM software, your Apple hardware works just like IBM hardware. Including your drives, monitors, printers, printer cards, clock cards and serial clocks.

You can use your IIe<sup>®</sup> or IIgs<sup>™</sup> keyboard with IBM software. Or use our optional IBM-style keyboard (required for the II Plus).

You can use your Apple mouse. Or an IBM compatible serial mouse.

#### **Plenty of power.**

PC Transporter gives you as much as 640K of user RAM and 128K of system RAM in the IBM mode.

PC Transporter also is an Apple expansion card, adding up to 768K of extra RAM in the Apple mode. The Apple expansion alone is a \$300 value.

#### **Easy to install.**

You can install PC Transporter in about 15 minutes, even if you've never added an expansion board. You don't need special tools. Simply plug it into an Apple expansion slot (1 through 7 except 3), connect a few cables and a disk drive, and go!



**PC Transporter taps into the world's largest software library. Now your Apple can run most of the IBM software you use at work. And it opens a new world of communications programs, games and bulletin boards.**

### A universal disk drive controller.

PC Transporter supports 3.5" and 5.25" MS-DOS and ProDOS formatted diskettes. You'll shift instantly between Apple ProDOS and IBM MS-DOS.

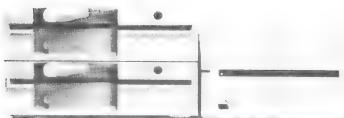
You'll need our versatile 5.25" 360K drive system to run IBM applications from 5.25" floppy disks. Use your Apple 5.25" drive for Apple 5.25" disks.

An Apple Disk 3.5 Drive will support the new 3.5" disks whether they're IBM MS-DOS formatted or Apple ProDOS formatted. The PC Transporter acts like an Apple Disk 3.5 Drive disk controller for IIgs, IIe, and II Plus users.

PC Transporter supports up to 5 drives in a number of combinations.

For example, you can connect a 5.25 Applied Engineering 360K dual-drive system directly to the card. Then plug two daisy-chained Apple 3.5 Drives (not the Apple UniDisk 3.5) to the dual-drive system. For a fifth drive, use a ProDOS file as an IBM hard disk.

**PC Transporter controls Apple and IBM compatible disk drives. It supports 3.5" and 5.25" MS-DOS and ProDOS formatted diskettes.**



**Versatile data storage.** PC Transporter reads MS-DOS and translates it into Apple native ProDOS. You can store IBM programs and data on any ProDOS storage device including the Apple 3.5 Drive, Apple UniDisk™ 3.5, Apple 5.25" drive, SCSI or ProDOS compatible hard drives. (You can use the Apple UniDisk 3.5 with its own controller card for storing programs and data, but not for directly booting an IBM formatted disk.)

You can even use our 360K PC compatible drive for ProDOS

### Make your Apple speak IBM.

#### PC Transporter memory choices.

RAM in Apple mode:	RAM in IBM mode:	Price:
<b>384K</b>	<b>256K</b>	<b>\$489.00</b>
<b>512K</b>	<b>384K</b>	<b>\$529.00</b>
<b>640K</b>	<b>512K</b>	<b>\$569.00</b>
<b>768K</b>	<b>640K</b>	<b>\$609.00</b>

Note: The IBM mode is 128K less because the PC Transporter uses 128K for system memory.

#### IIgs Installation Kit 49.00

#### IIe/II Plus Installation Kit 39.00

#### PC Transporter Accessories

##### 5.25" IBM Format 360K

##### Drive Systems

**Single-Drive System** 269.00

**Dual-Drive System** 399.00

**Half-Height Drive** 135.00

(add to Single-Drive system to make Dual-Drive)

**IBM-Style Keyboard** 139.00  
(required for Apple II Plus. Requires IBM Keyboard Cable.)

**IBM Keyboard Cable** 34.00

**Sony RGB Monitor** 499.00

**Analog RGB Cable** 39.00

(for use with Sony monitor)

**Digital RGB Cable** 39.00

(for use with Sony monitor)

**Digital RGB Adapter** 24.00

**ColorSwitch** 44.00

(included with IIgs Installation Kit)

**128K ZIP** 40.00

**PC Transporter Memory Expansion** per set

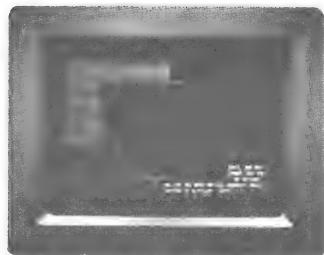
**Chip Set** 8087-2 Math

**Co-processor Chip** 229.00

**Heavy Duty Power Supply** 69.00

(IIe and II Plus only)

See your dealer or call or send check or money order to Applied Engineering. MasterCard, VISA and COD welcome. Texas residents add 6 1/4% sales tax.



**PC Transporter produces better IBM graphics than IBM Analog is sharper than digital. So with an analog RGB monitor, PC Transporter's CGA graphics and text are superior to IBM's digital display — even while running IBM software!**

*And, you can also use an Apple composite monitor in IBM text or graphics mode.*

storage and a 143K Apple 5.25" drive for MS-DOS storage.

### Created by Apple's original designers.

The brains behind PC Transporter were also behind your Apple II.

The PC Transporter design team includes the former project managers for the creation of the Apple IIe and IIc. The co-designer of the Apple II disk controller. And the first full-time Apple programmer and author of the ProDOS operating system.

So you know the PC Transporter and your Apple were made for each other.

### Support and service from the leader in Apple add-ons.

Applied Engineering sells more Apple peripheral boards than anyone else — including Apple Computer. So you know we'll be around after the sale.

PC Transporter comes with a 15-day money back guarantee. If you're not fully satisfied after using it, return it for a full refund. PC Transporter also comes with a 1-year warranty.

### How to get your PC Transporter today.

See your dealer. Or call Applied Engineering any day between 9 a.m. and 11 p.m. CST at 214-241-6060.

**AE Applied Engineering**  
*The Apple enhancement experts.*

PO. Box 798, Carrollton, TX 75006

214-241-6060

A Division of AE Research Corporation

```

00-          1010 HAVE.PAD .EQ 0   =0 IF NO PAD, =1 IF PAD PRESENT
1020          .OP 65816
1030 *-----*
1040 #SAVE S.KEY.EDIT
1050 *-----*
1060 .OR $5000
1070 .TF KEY.EDIT
1080 *-----*
1090 * VARIABLES NEEDED FOR LOADING
1100 * KEY.EDIT
1110 *-----*
03EA-        1120 DOS.IO.HOOK .EQ $3EA
A7D4-        1130 BLD.DOS.BUFRS .EQ $47D4
9D00-        1140 DOS.BUFR.LOC .EQ $9D00
38-          1150 KSWL .EQ $38
39-          1160 KSWH .EQ $39
7300-        1170 COL.80 .EQ $C300
3E22-        1180 MON.RESET .EQ $3E2
03F4-        1190 PWRUP .EQ $3F4
1200 *-----*
005000- A9 9B 1210 BGN LDA /RESET.PTCH
005002- CD F3 03 1220 CMP MON.RESET+1
005005- FO 47 1230 BEQ .6 ALREADY SETUP
1240 ***Copy BODY to $9900-9BFF-----
005007- A0 00 1250 LDY #0
005009- B9 87 50 1260 .1 LDA IMAGE,Y
00500C- 99 00 99 1270 STA $9900,Y
00500B- B9 87 51 1280 LDA IMAGE+256,Y
005012- 99 00 9A 1290 STA $9A00,Y
005015- B9 87 52 1300 LDA IMAGE+512,Y
005016- 99 00 9B 1310 STA $9B00,Y
00501B- C8      1320 INY
00501C- D0 EB 1330 BNE .1
1340 ***Clear command buffer-----
1350 JSR CLEAR.BUFFER
1360 ***Set up RESET vector-----
005021- A0 01 1370 LDY #1
005023- B9 F2 03 1380 .5 LDA MON.RESET,Y POINT MY RESET AT CURRENT
005026- 99 9C 9B 1390 STA NORM.RESET,Y
005029- B9 85 50 1400 LDA MY.RESET,Y POINT RESET AT MY PATCH
00502C- 99 00 9B 1410 STA MON.RESET,Y
00502F- 88      1420 DEY
005030- 10 F1 1430 BPL .5
005032- A9 3E 1440 LDA /RESET.PTCH+$A500 VALIDATE THE VECTOR
005034- 8D F4 03 1450 STA PWRUP
1460 ***Drop DOS buffers 4 pages-----
1470 SEC
1480 LDA DOS.BUFR.LOC+1
1490 SBC #4
1500 STA DOS.BUFR.LOC+1
1510 JSR BLD.DOS.BUFRS
1520 ***Install my input hook-----
005037- 38 1530 LDA #HOOK
005038- AD 01 9D 1540 LDY /HOOK
00503B- E9 04 1550 STA KSWL
00503D- 8D 01 9D 1560 STY KSWH
005040- 20 D4 A7 1570 JSR DOS.IO.HOOK
00504E- 60      1580 .6 RTS
1590 *-----*
1600 T
1610 ***Copy BODY to $9900-9BFF-----
1620 LDY #0
1630 .1 LDA IMAGE,Y
005054- 99 00 99 1640 STA $9900,Y
005057- B9 87 51 1650 LDA IMAGE+256,Y
00505A- 99 00 9A 1660 STA $9A00,Y
00505D- B9 87 52 1670 LDA IMAGE+512,Y
005060- 99 00 9B 1680 STA $9B00,Y
005063- C8      1690 INY
005064- D0 EB 1700 BNE .1
1710 ***Clear command buffer-----
1720 JSR CLEAR.BUFFER
1730 ***Setup RESET Vector-----
005069- A0 01 1740 LDY #1
00506B- B9 85 50 1750 .5 LDA MY.RESET,Y POINT RESET AT MY PATCH
00506E- 99 F2 03 1760 STA MON.RESET,Y
005071- 88      1770 DEY
005072- 10 F7 1780 BPL .5
005074- A9 3E 1790 LDA /RESET.PTCH+$A500 VALIDATE THE VECTOR
005076- 8D F4 03 1800 STA PWRUP
1810 ***Install my input hook-----
005079- A9 00 1820 LDA #HOOK
00507B- A0 99 1830 LDY /HOOK
00507D- 85 38 1840 STA KSWL
00507F- B4 39 1850 STY KSWH
005081- 20 EA 03 1860 JSR DOS.IO.HOOK
005084- 60      1870 .6 RTS
1880 *-----*

```

005085- 8A 9B 1890 MY.RESET .DA RESET.PTCH  
 1900 \*-----  
 1910 \* VARIABLES AND CONSTANTS  
 1920 \*-----  
 FBF4- 1930 MON\_ADVANCE .EQ \$FBF4  
 FC22- 1940 MON\_VTAB .EQ \$FC22  
 FD0C- 1950 MON\_RDKEY .EQ \$FD0C  
 FC42- 1960 MON\_CLREOP .EQ \$FC42  
 FD2F- 1970 MON\_ESC .EQ \$FD2F  
 FDED- 1980 MON\_COUT .EQ \$FDED  
 1990 \*-----  
 FD1B- 2000 KEYIN.40 .EQ \$FD1B  
 C305- 2010 KEYIN.80 .EQ \$C305  
 2020 \*-----  
 0200- 2030 INBUF .EQ \$200  
 C01R- 2040 COL.STATE .EQ \$C01F  
 C025- 2050 KEY.STATE .EQ \$C025  
 25- 2060 CV .EQ \$25  
 24- 2070 CH40 .EQ \$24  
 057B- 2080 CH80 .EQ \$57B  
 23- 2090 WNDWDTH .EQ \$21  
 2100 WNDBTM .EQ \$23  
 2110 \*-----  
 2120 IMAGE .PH \$9900  
 2130 \*-----  
 2140 \* The input hook at KSWL,H branches here whenever  
 2150 \* RDKEY is called.  
 2160 \*-----  
 009900- 80 03 2170 HOOK BRA .2 <<<MODIFIED TO SKIP OR NOT SKIP  
 009902- 4C 68 9B 2180 .1 JMP TRUE.KEYIN THIS JMP>>>  
 009905- EC A1 9B 2190 .2 CPX LNGTH IS X POSITION GREATER THAN MY SAVED LENGTH?  
 009908- 90 02 2200 BCC .3 NO. MUST BELONG TO ME  
 00990A- DO F6 2210 BNE .1 IF NOT EQUAL, THEN IT IS NOT MINE  
 2220 \*-----  
 00990C- 48 2230 .3 PHA SAVE THE CURRENT SCRN CHAR  
 00990D- 8A 2240 TXA AT BEGINNING OF LINE?  
 00990E- DO 12 2250 BNE .5 NO.  
 2260 \*----Save line start position----  
 009910- A5 25 2270 LDA CV GET POSITION OF LINE START AND SAVE  
 009912- 8D A0 9B 2280 STA BOL  
 009915- A5 24 2290 STA BOL  
 009917- 2C 1F CO 2300 LDA CH40 ASSUME NO COLUMNS  
 00991A- 10 03 2310 BIT COL.STATE IS IT 40 OR 80 COLS?  
 00991C- AD 7B C5 2320 BPL .4 ...40 COLUMNS  
 00991F- 8D 9F 9B 2330 LDA CH80 ...80 COLUMNS  
 009922- 68 2340 .4 STA BOC  
 009923- 20 68 9B 2350 PLA RETRIEVE SAVED SCREEN CHARACTER  
 009926- 9C 01 99 2360 JSR TRUE.KEYIN GET A CHR FROM THE NORMAL INPUT ROUTINE  
 009929- 20 31 99 2370 STZ HOOK+1 SWITCH TO LET MON.RDKEY FUNCTION  
 00992C- 20 0C FD 2380 JSR PROCESS.CHAR  
 00992F- 80 F8 2390 JSR MON.RDKEY  
 2400 BRA .6 NORMAL CHARS BUST THE LOOP  
 2410 \*-----  
 009931- 8D A7 9B 2420 PROCESS.CHAR STA CURRCHAR  
 009934- A0 FC 2430 LDY #4  
 009936- C8 2440 .1 INY  
 009937- C8 2450 INY  
 009938- C8 2460 INY  
 009939- C8 2470 INY  
 00993A- B9 57 99 2480 LDA CMDTBL,Y  
 00993D- F0 0F 2490 BEQ .2 ...END OF CMDTBL  
 00993E- CD A7 9B 2500 CMP CURRCHAR  
 009942- DO F2 2510 BNE .1 ...TRY NEXT ENTRY  
 009944- AD 25 CO 2520 LDA KEY.STATE  
 009947- 29 D3 2530 AND #\$11010011 ONLY OA,SA,PAD,CTRL,SHIFT  
 009949- D9 58 99 2540 CMP CMDTBL+1,Y  
 00994C- DO E8 2550 BNE .1 ...TRY NEXT ENTRY  
 00994E- B9 5A 99 2560 .2 LDA CMDTBL+3,Y  
 009951- 48 2570 PHA  
 009952- B9 59 99 2580 LDA CMDTBL+2,Y  
 009955- 48 2590 PHA  
 009956- 60 2600 RTS  
 2610 \*-----  
 2620 .MA CMD  
 2630 .DA #\$11,\$\$12,\$3-1  
 2640 .EM  
 2650 \*-----  
 2660 CMDTBL  
 009957- 88 00 96 99 2670 >CMD 88,00,BAKSPC LEFT ARROW  
 009958- 0000 >DA #\$88,\$00,BAKSPC-1  
 009958- 88 80 F2 99 2680 >CMD 88,80,LINE.START OA-LEFT ARROW  
 009958- 0000 >DA #\$88,\$\$80,LINE.START-1  
 009958- 95 00 A0 99 2690 >CMD 95,00,FORWD RIGHT ARROW  
 009958- 0000 >DA #\$95,\$\$00,FORWD-1  
 009958- 95 80 F9 9A 2700 >CMD 95,80,END.OF.LINE OA-RIGHT ARROW  
 009958- 0000 >DA #\$95,\$\$80,END.OF.LINE-1  
 009967- 2710 >CMD FF,00,DELCHR DELETE  
 009967- FF 00 B0 99 2710 .DA \$\$FF,\$\$00,DELCHR-1

```

00996B- FF 80 F7 99 0000> 2720 >CMD FF 80 DELALL OA-DELETE
00996B- FF 80 F7 99 0000> 2730 >CMD FF 02 DELEOL CTRL-DELETE
00996F- FF 02 FA 99 0000> .DA #$FF,$#02,DELEOL-1
009973- FF 03 D5 99 0000> 2740 >CMD FF 03 DELBOL CTRL-SHIFT-DELETE
009973- FF 03 D5 99 0000> .DA #$FF,$#03,DELBOL-1
009977- FF 40 2C 98 0000> 2750 >CMD FF ,40 CLEAR.BUFFER SA-DELETE
009977- FF 40 2C 98 0000> .DA #$FF,$#40,CLEAR.BUFFER-1
009977- FF 40 2C 98 0000> 2760 .DO HAVE PAD
009977- FF 40 2C 98 0000> 2770 >CMD AE ,90 DELCUR OA-PAD-"
009977- FF 40 2C 98 0000> 2780 >CMD BO ,90 INS.TOG OA-PAD-0"
009977- FF 40 2C 98 0000> 2790 .ELSE
009977- FF 40 2C 98 0000> 2800 >CMD AE ,80 DELCUR OA-"
00997B- AE 80 B3 99 0000> 2810 >CMD AC ,80 INS.TOG OA-"
00997F- AC 80 00 9A 0000> 2820 .FIN
009983- 8B 00 33 9A 0000> 2830 >CMD 8B ,00 UP UP ARROW
009983- 8B 00 33 9A 0000> 2840 >CMD 8A ,00 DOWN DOWN ARROW
009987- 8A 00 18 9A 0000> 2850 >CMD 89 ,00 TAB.FWD TAB
00998B- 89 00 6C 9A 0000> 2860 >CMD 89 ,80 TAB.BAK OA-TAB
00998P- 89 80 7A 9A 0000> 2870 >CMD 00 ,00 NORM.CHR any other
009993- 00 00 85 9A 0000> 2880 .DA #$00,$#00,NORM.CHR-1
009997- 8A 00 00 00 0000> 2890 BAKSPC TXA AT LINE START?
009998- F0 06 2900 BEQ RTS.1 YES, GET THE NEXT CHR
00999A- CA 00 00 00 0000> 2910 DEX NO, BACKUP ONE SPACE
00999B- A9 88 2920 LDA #$88 PRINT A BACKSPACE
00999D- 4C ED FD 2930 COUT.1 JMP MON.COUT
0099A0- 60 2940 RTS.1 RTS
0099A1- EC A1 9B 2950 *
0099A4- B0 FA 2960 FORMD CPX LNGTH ALREADY AT END OF LINE?
0099A6- EB 2970 BCS RTS.1 ...YES
0099A7- A9 9C 2980 INX ...NO, ADVANCE
0099A9- 2C 1F CO 2990 LDA #$9C PRINT $9C TO SPACE FORWARD
0099AC- 30 EF 3000 BIT COL.STATE 80- OR 40-COLUMNS?
0099AE- 4C F4 FB 3010 BMI COUT.1 ...80-COLUMNS
0099BD- 20 97 99 3020 JMP MON.ADVANCE ...40-COLUMNS
0099B1- 20 42 FC 3030 DELCHR JSR BAKSPC Delete char to left of cursor
0099B4- 20 42 FC 3050 DELCUR JSR MON.CLREOP Delete char under cursor
0099B7- AD A1 9B 3060 LDA LNGTH
0099BA- F0 19 3070 BEQ .2
0099BC- DA 3080 PHX
0099BD- E8 3090 .1 INX
0099BE- EC A1 9B 3100 CPX LNGTH
0099C1- B0 0B 3110 BCS .3
0099C3- BD 00 02 3120 LDA INBUF,X MOVE INBUF DOWN BY ONE
0099C6- 9F FF 01 3130 STA INBUF-1,X
0099C9- 20 ED FD 3140 JSR MON.COUT
0099CC- 80 EF 3150 BRA .1
0099CE- FA 3160 .3 PLX RESTORE CURSOR POSITION ON SCREEN
0099CF- 20 FD 9A 3170 JSR CURSOR.POSN
0099D2- CE A1 9B 3180 DEC LNGTH
0099D5- 60 3190 .2 RTS
0099D6- EC A1 9B 3200 *
0099D9- B0 1D 3210 DELBOL CPX LNGTH If at eol, delete entire line
0099DB- DA 3220 BCS DEALL
0099DC- 20 F3 99 3230 PHX SAVE LOCAL POSITION WITHIN INBUF
0099DF- 7A 3240 JSR LINE.START GO TO BEGINNING OF LINE
0099E0- B9 00 02 3250 PLY (Y) points at remaining chars
0099E3- DD 00 02 3260 .1 LDA INBUF,Y MOVE INBUF DOWN TO BEGINNING OF BUFFER
0099E8- 20 ED FD 3270 STA INBUF,X AND WRITE TO SCREEN
0099E9- C8 3280 JSR MON.COUT
0099EA- E8 3290 INY
0099EB- CC A1 9B 3300 INX
0099EE- 90 F0 3310 CPY LNGTH
0099F0- 20 FB 99 3320 BCC .1
0099F3- A2 00 3330 JSR DEEOL LOP OFF THE REST
0099F5- 4C FD 9A 3340 LINE.START INDICATE BEGINNING OF INBUF
0099F5- 4C FD 9A 3350 LDY #0
0099F5- 4C FD 9A 3360 JMP CURSOR.POSN
0099F8- 20 F3 99 3370 *
0099FB- 8E A1 9B 3380 DELALL JSR LINE.START Delete entire line
0099FB- 8E A1 9B 3390 DEEOL STX LNGTH Delete from cursor to eol
0099FE- 4C 42 FC 3400 JMP MON.CLREOP CLEAR TO END OF WINDOW
0099FE- 4C 42 FC 3410 *
009A01- AC A6 9B 3420 INS.TOG
009A04- AF 34 01 E1 3430 LDY CURSOR SWAP THE CURSORS
009A04- AF 34 01 E1 3440 LDA $E10134 CURRENT ACTIVE CURSOR
009A08- 8D A6 9B 3450 STA CURSOR SAVE IT
009A0B- 8B 3450 TYA PREVIOUS CURSOR
009A0C- 8F 34 01 E1 3460 STA $E10134 START USING IT AGAIN
009A10- AD A2 9B 3470 LDA INS.FLAG TOGGLE THE FLAG

```

```

009A13- 49 80      3490    EOR #$80
009A15- 8D A2 9B    3500    STA INS.FLAG
009A18- 60          3510    RTS
009A19- 20 4F 9A    3520    *
009A1C- 10 3D      3530    * Select stored input line from buffer
009A1E- 88          3540    by scanning forward in time
009A1F- C8          3550    *
009A20- B9 00 9C    3560    DOWN JSR PREPARE.BUFFER.SEARCH
009A23- D0 FA      3570    BPL RTS.2 Buffer is empty
009A25- C8          3580    DEY
009A26- B9 00 9C    3590    .1 INY SEARCH FOR "00"
009A29- F0 FA      3600    LDA BUFFER,Y
009A2B- 20 5C 9A    3610    BNE .1
009A2B- 8C A5 9B    3620    INY SEARCH FOR NON-ZERO
009A31- 4C FD 9A    3630    LDA BUFFER,Y
009A32-             3640    BEQ .2
009A32-             3650    JSR CBTB.1 STORE CHAR AND COPY REST OF CMND
009A32-             3660    STY WHERE
009A32-             3670    JMP CURSOR.POSN RTN WITH CURSOR AT LINE END, CHK ADJUSTMENT
009A32-             3680    *
009A32-             3690    * Select stored input line from buffer
009A32-             3700    by scanning backward in time.
009A32-             3710    *
009A34- 20 4F 9A    3720    UP JSR PREPARE.BUFFER.SEARCH
009A37- 10 22      3730    BPL RTS.2 Buffer is empty
009A39- C8          3740    INY
009A39- 88          3750    .1 DEY BACKUP TO NON-ZERO
009A3B- B9 00 9C    3760    LDA BUFFER,Y
009A3B- F0 FA      3770    BEQ .1
009A40- 88          3780    .2 DEY BACKUP TO "00"
009A41- B9 00 9C    3790    LDA BUFFER,Y
009A41- D0 FA      3800    BNE .2
009A46- 8C A5 9B    3810    STY WHERE
009A49- 20 63 9A    3820    JSR CBTB.2 COPY COMMAND TO INBUF
009A4C- 4C FD 9A    3830    JMR CURSOR.POSN RTN WITH CURSOR AT LINE END, CHK ADJUSTMENT
009A4C-             3840    *
009A4C-             3850    PREPARE.BUFFER.SEARCH
009A4F- 20 F3 99    3860    JSR LINE.START GO TO BEGINNING OF LINE
009A52- 20 F3 FC    3870    JSR MON.CLRCP CLEAR THE LINE
009A55- AC A5 9B    3880    LDY WHERE GET LAST POSITION IN BUFFER
009A58- 2C A3 9B    3890    BIT BUF.FLAG ANYTHING IN BUFFER?
009A58- 60          3900    RTS.2 RTS
009A5C- 9D 00 02      3910    *
009A5F- 20 ED FD    3920    CBTB.1 STA INBUF,X
009A62- 28          3930    JSR MON.COUT
009A62-             3940    INX
009A63- C8          3950    CBTB.2 INY COPY BUFFER TO INPUT BUFFER AND
009A63-             3960    LDA BUFFER,Y DISPLAY ON SCREEN
009A67- B9 00 9C    3970    BNE CBTB.1
009A67- D0 F3        3980    STX LNGTH SAVE TOTAL LINE LENGTH
009A69- BE A1 9B    3990    RTS
009A6C- 60          4000    *
009A6D- EC A1 9B    4010    TAB.FWD
009A70- B0 06      4020    .1 CPX LNGTH ELSE, MOVE FORWARD IF NOT AT LINE END
009A72- E8          4030    BCS NEWPOS
009A73-             4040    INX
009A73- 20 76 9B    4050    JSR COMPARE.TAB.CHARS
009A73- 90 F5        4060    BCC .1 NO. GET THE NEXT INBUF CHAR
009A78- 4C FD 9A    4070    NEWPOS JMP CURSOR.POSN YES. CALC NEW POSITION OF CURSOR
009A78-             4080    *
009A7B- 8A          4090    TAB.BAK
009A7C- F0 FA        4100    .1 TXA TAB BACKWARD IF NOT AT LINE BEGINNING
009A7E- CA          4110    BEQ NEWPOS
009A7F-             4120    DEX
009A7F- 20 76 9B    4130    JSR COMPARE.TAB.CHARS
009A82- 90 F7        4140    BCC .1
009A84- B0 F2        4150    BCS NEWPOS
009A84-             4160    *
009A86- 68          4170    NORM.CHR
009A87- 68          4180    PLA POP A RETURN ADDRESS
009A88- 20 27 9B    4190    PLA
009A88- AD A7 9B    4200    JSR DRCT.OFF
009A8E- 48          4210    LDA CURRCHAR GET INPUT CHAR
009A8E-             4220    PHA SAVE CHR FOR LATER CODE
009A8F- C9 A0        4230    CMP #$A0 IS IT A CONTROL CHAR?
009A91- B0 31        4240    BCS .2 ...NO
009A93- C9 8D        4250    CMP #$BD CARRIAGE RETURN?
009A95- D0 06        4260    BNE .0
009A97- 20 3A 9B    4270    JSR MOVE TO BUFFER
009A9A- 20 FA 9A    4280    JSR END.OF.LINE
009A9D- 2C A2 9B    4290    .0 BIT INS.FLAG INSERTION MODE ON?
009AA0- 10 03        4300    BPL .1 NO.
009AA2- 20 01 9A    4310    JSR INS.TOG YES. TOGGLE INSERT MODE OFF
009AA5- 9C A1 9B    4320    .1 STZ LNGTH CLEAR TOTAL LINE LENGTH
009AA8- 68          4330    PLA
009AA9- C9 9B        4340    CMP #$9B ESC CHARACTER?
009AAB- DO 2D        4350    BNE .5

```

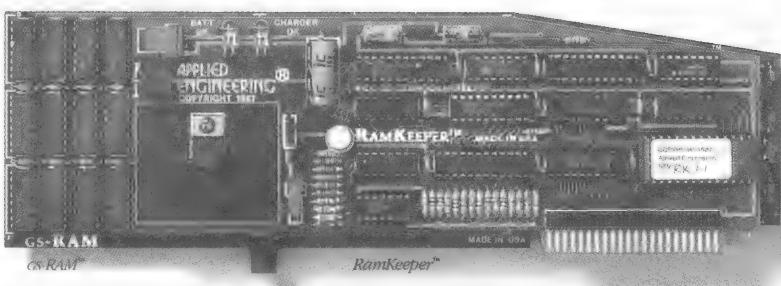
```

009AAD- 9C 01 99 4360 ----Handle ESC-----
009AB0- 20 2F FD 4370 STZ HOOK+1 SET CODE FOR PASS THRU WHILE IN 'ESC' MODE
009AB3- 48 4380 JSR MON.ESC LET MONITOR HANDLE ESCAPE MOVES
009AB4- 20 27 9B 4390 PHA SAVE CHR ON THE STACK
009AB7- 08 25 C0 4400 JSR DRCT.OFF RESET SET CODE TO CHK EACH CHR
009AB8- 80 1D 4410 ASL KEY.STATE MOVE OPEN APPLE STATUS TO CARRY
009ABC- A9 A0 4420 BCS .4 AND RTN IF SET
009ABE- 9D 00 02 4430 LDA # " ELSE, INSERT A SPACE INTO INBUF
009ACT- E8 4440 STA INBUF,X
009AC1- E8 4450 INX AND INCREMENT POSITION SO THAT NEXT TIME THRU
009AC2- 80 15 4460 * KEY.EDIT WILL IGNORE THE LINE
009AC4- 0E 25 C0 4470 BRA .4
009AC7- B0 10 4480 .2 ASL KEY.STATE MOVE STATUS OF OPEN APPLE KEY TO CARRY
009AC9- 2C A2 9B 4490 BCS .4 IF SET, THEN RTN NOW
009ACC- 3D OD 4500 BIT INS.FLAG INSERTION MODE ON?
009ACD- EC A1 9B 4510 BMI INS.CHR YES. GO HANDLE IT
009AD1- 90 06 4520 CPX LNGTH NO. INC LENGTH IF AT END.
009AD3- 20 FD 9A 4530 BCC .4
009AD6- EE A1 9B 4540 JSR CURSOR.POSN POSITION CURSOR AT LINE END AND CHK
009AD9- 68 4550 INC LNGTH ADJUSTMENT FOR BTM OF WINDOW
009ADA- 60 4560 PLA GET CHAR FROM STACK AND RTN
009ADA- 60 4570 RTS
009ADA- 60 4590
4600 * This portion handles character insertions
4610 * while the insert flag is on.
4620 *
4630 INS.CHR
009ADB- 7A 4640 PLY GET CHR FROM STACK INTO Y-REG
009ADC- 5A 4650 PHY LEAVE ON STACK TOO
009ADD- DA 4660 PHX SAVE LOCAL POSITION WITHIN INBUF
009ADE- EE A1 9B 4670 INC LNGTH INCREASE LINE LENGTH BY ONE
009AE1- 98 4680 .1 TYA INSERT CHAR IN INBUF
009AE2- BC 00 02 4690 LDY INBUF,X GET CURRENT CHAR
009AE5- 9D 00 02 4700 STA INBUF,X PUT NEW CHAR
009AE8- 20 ED FD 4710 JSR MON.COUT AND DISPLAY ON SCREEN
009AEC- E8 4720 IMX MOVE ON DOWN THE LINE
009AEC- EC A1 9B 4730 CPX LNGTH MORE TO GO...
009AEF- 90 F0 4740 BCC .1 ADJUSTMENT NEEDED FOR BEING NEAR WINDOW BTM?
009AF1- 20 FD 9A 4750 JSR CURSOR.POSN RESET POSITION IN INBUF
009AF4- FA 4760 PLX RESET CURSOR TO ITS ORIGINAL POSITION
009AF5- 20 FD 9A 4770 JSR CURSOR.POSN INSERTED CHARACTER
009AF8- 68 4780 PLA
009AF9- 60 4790 RTS
009AF9- 60 4800
4810 END.OF.LINE
009AFA- AE A1 9B 4820 LDX LNGTH CALCULATE OFFSET FROM LINE START
4830 *
4840 * (I)=position in INBUF
4850 * Compute screen line and column for current position
4860 * and position cursor there.
4870 * If that is below window, adjust BOL accordingly and
4880 * position to bottom line.
4890 *
4900 CURSOR.POSN
009AFD- AC A0 9B 4910 LDY BOL GET ROW OF LINE START
009B00- 18 4920 CLC virtual screen position = BOC+X
009B01- 8A 4930 TXA
009B02- 6D 9F 9B 4940 ADC BOC
009B05- C5 21 4950 *----Adjust for window width-----
009B07- 90 05 4960 .1 CMP WNDWDTH
009B09- E5 21 4970 BCC .2 THIS IS THE LINE
009B0B- C8 4980 SBC WNDWDTH
009B0C- 80 F7 4990 INY MOVE DOWN ONE LINE
009B0C- 80 F7 5000 BRA .1
009B0E- 8D 7B 05 5010 *----HTAB to position-----
009B11- 2C 1F C0 5020 .2 STA CH80
009B14- 30 02 5030 BIT COL.STATE In 80-column mode?
009B16- 85 24 5040 BMI .3 ...yes
009B16- 85 24 5050 STA CH40 ...no, store in 40-col CH
009B18- C4 23 5060 *----Adjust if below window-----
009B1A- 90 06 5070 .3 CPY WNDBTM
009B1C- C8 A0 9B 5080 BCC .4 ON THE SCREEN NOW
009B1F- 88 5090 DEC BOL ADJUST BEGINNING OF LINE ROW NUMBER
009B20- D0 F6 5100 DEY
009B20- D0 F6 5110 BNE .3
009B22- 84 25 5120 *----VTAB to line-----
009B24- 4C 22 FC 5130 .4 STY CV
009B24- 4C 22 FC 5140 JMP MON.VTAB SET NEW LINE ROW VALUE
009B27- A9 03 5150 *
009B29- 8D 01 99 5160 DRCT.OFF
009B2C- 60 5170 LDA #3
009B2C- 60 5180 STA HOOK+1
009B2C- 60 5190 RTS
009B2C- 60 5200 *

```

# RamKeeper™

For the "Instant On" Apple IIgs.



## Permanent Storage with an "Electronic Hard Disk"

Now when you turn on your IIgs your favorite program can appear on screen in just a few seconds! With RamKeeper, your IIgs memory card will retain stored programs and stored data while your IIgs is turned off. RamKeeper allows you to divide your IIgs memory into part "electronic hard disk" and part RAM for your programs workspace—in almost any way you want and at anytime you want. Gs-RAM, Gs-RAM Plus, Apple IIgs memory card and most other IIgs memory cards are compatible with RamKeeper.

## Supports Up to Two IIgs Memory Cards at the Same Time

If you bought your IIgs with Apple's memory card and later wished you had the Gs-RAM, no problem. RamKeeper will support both cards plugged into RamKeeper simultaneously!

## How it Works

Just unplug your IIgs memory card



*"I've purchased several Applied Engineering products over the years. They're always well made and performed as advertised. I recommend them wholeheartedly."*

*Steve Wozniak, the creator of Apple Computer*

from your computer, plug your IIgs memory card into RamKeeper, plug RamKeeper into the IIgs memory slot. If you have another IIgs memory card, an additional card socket on RamKeeper will accommodate your second card. That's all there is to it!

## Reliability from the Most Experienced

Applied Engineering has the most experience in the industry with battery-backed memory for the Apple so you are assured of the most reliable memory back-up system available. And in the world of battery-backed memory, reliability is everything! That's why Applied Engineering uses the more dependable Gel-Cell's instead of Ni-Cad batteries (if Ni-Cad's aren't discharged periodically, they lose much of their capacity). RamKeeper has close to 6 times (about 6 hours) the "total power failure" back-up time of other systems. When power returns, RamKeeper automatically re-charges the battery to a full charge. With power from your wall outlet, RamKeeper will back-up your IIgs memory cards RAM indefinitely.

## RamKeeper Has and Does It All!

- Allows instant access to your programs without slow disk delays
- Configure Kilobytes or Megabytes of instant ROM storage for your favorite programs

- Reduces power strain to your internal IIgs' power supply
- Contains back-up status LED's
- Can support up to two IIgs memory cards simultaneously
- Supports both 256K installed memory chip boards like Gs-RAM and the Apple IIgs Memory Expansion Card as well as 1 MEG installed memory chip boards like Gs-RAM Plus
- 5-year hassle-free warranty
- 15 day money back guarantee
- Proudly made in the U.S.A.
- RamKeeper comes complete with battery, software and documentation

## Only \$179.00!

*(Gs-RAM card shown in photo not included)*

## Order Your RamKeeper Today!

See your dealer or call (214) 241-6060, 9:00-11:00 CST, 7 days a week, or send check or money order to Applied Engineering MasterCard, VISA and C.O.D. welcome. Texas residents add 7% sales tax. Add \$10.00 if outside U.S.A.

**A** APPLIED ENGINEERING™

*The Apple enhancement experts.*

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

*Prices subject to change without notice.*

```

009B2D- A0 00 5210 CLEAR.BUFFER      ZERO CONTENTS OF STORAGE BUFFER
009B2F- BB 5220 LDY #0
009B30- 99 00 9C 5230 TYA
009B31- C8 5240 .1 STA BUFFER,Y
009B32- DO FA 5250 INY
009B33- 8D A3 9B 5260 BNE .1
009B34- 8D A3 9B 5270 STA BUF.FLAG
009B35- 60 5280 RTS
009B39- 60 5290 *
5300 MOVE TO BUFFER
009B3A- AD A1 9B 5310 LDA LNGTH ANY CHARACTERS IN INBUF?
009B3D- F0 28 5320 BEQ .3 ...NO, RETURN NOW
009B3F- DA 5330 PHX YES. SAVE POSITION WITHIN INBUF
009B40- AC A4 9B 5340 LDY TOP MOVE INBUF TO STORAGE BUFFER
009B43- A2 00 5350 LDX #0
009B45- C8 5360 .1 INY POINT TO NEXT LOCATION IN BUFFER
009B46- BD 00 02 5370 LDA INBUF,X MOVE INBUF AND PLACE ON TOP
009B49- 99 00 9C 5380 STA BUFFER,Y
009B4C- E8 5390 INX
009B4D- EC A1 9B 5400 CPX LNGTH
009B50- 90 F3 5410 BCC .1
009B52- 8D A3 9B 5420 STA BUF.FLAG TURN BUFFER FLAG ON
009B55- C8 5430 INY
009B56- 8C A5 9B 5440 STY TOP MARK NEW POSITION OF TOP
009B59- 8C A5 9B 5450 STY WHERE AND WHERE WE START AGAIN
009B5C- BB 5460 TVX
009B5D- 9E 00 9C 5470 .2 STZ BUFFER,X ZERO OUT ANY RESIDUAL CMNDS
009B60- E8 5480 INX
009B61- BD 00 9C 5490 LDA BUFFER,X
009B64- DO F7 5500 BNE .2
009B66- FA 5510 PLX
009B67- 60 5520 .3 RTS
5530 *
5540 TRUE.KEYIN
009B68- 0E 1F CO 5550 ASL COL.STATE 40- OR 80-COLUMNS?
009B6B- 6E 9E 9B 5560 ROR KYBRD SAVE ANSWER IN KEYBOARD STORAGE BYTE
009B6E- 30 03 5570 BMI .1 ...80
009B70- 4C 1B FD 5580 JMP KEYIN.40
009B73- 4C 05 C3 5590 .1 JMP KEYIN.80
5600 *
5610 COMPARE.TAB.CHARS
009B76- BD 00 02 5620 LDA INBUF,X GET CURRENT CHAR FROM LINE
009B79- A0 04 5630 LDY #TAB.SZ-1 NUMBER OF TAB CHARACTERS
009B7B- D9 85 9B 5640 .1 CMP TAB.CHARS,Y IF THEY ARE THE SAME, RTN WITH CARRY SET
009B7E- F0 04 5650 BEQ .2 ELSE GO CHK THE NEXT CHAR
009B80- 88 5660 DEY ...MORE IN LIST
009B81- 10 F8 5670 BPL .1
009B83- 18 5680 CLC NO TAB CHARACTERS MATCH SO CLEAR CARRY AND
009B84- 60 5690 .2 RTS RETURN TO CALLER
5700 *
009B85- A0 AC AE BB 5710 TAB.CHARS .AS -" ":
009B89- BA 5720 TAB.SZ .EQ #TAB.CHARS
05- 5730 *
5740 * COMES HERE DURING PROCESSING OF "RESET"
5750 *
5760 RESET.PTCH
009B8A- 20 27 9B 5770 JSR DRCT.OFF
009B8D- 2C 9E 9B 5780 BIT KYBRD WAS I IN 80-COLUMN?
009B90- 10 03 5790 BPL .1 ...NO
009B92- 20 00 C3 5800 JSR COL80 ...YES
009B95- 64 38 5810 .1 STZ KSWL HOOK MYSELF IN
009B97- A9 99 5820 LDA /HOOK
009B99- 82 39 5830 STA KSWH
009B9B- 4C DO 03 5840 JMP $3D0 FILLED IN BY INIT CODE
9B9C- 5850 NORM.RESET .EQ #-2
5860 *
009B9E- 00 5870 KYBRD .DA #0
009B9F- 5880 BOC .BS 1
009BA0- 5890 BOL .BS 1
009BA1- 00 5900 LNGTH .DA #0
009BA2- 00 5910 INS.FLAG .DA #0
009BA3- 00 5920 BUF.FLAG .DA #0
009BA4- 00 5930 TOP .DA #0
009BA5- 00 5940 WHERE .DA #0
009BA6- DE 5950 CURSOR .AS -"/"
009BA7- 5960 CURRCHAR .BS 1
5970 *
5980 DO #>$9BFF
5990 ...ERROR: KEY.EDIT IS LONGER THAN 3 PAGES...
6000 .ELSE
6010 BUFFER .EQ $9C00
6020 .FIN
6030 *
6040 .EP

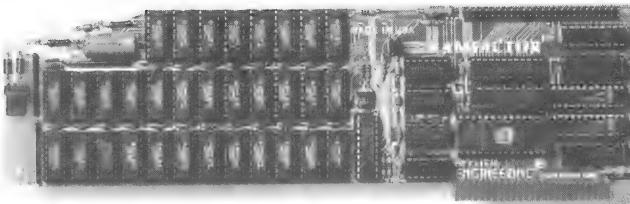
```

0000 ERRORS IN ASSEMBLY

# RamFactor™

*Boot any II+, IIe, or IIgs and be up and running your favorite software in less than 1 second.*

Now anyone with an Apple II+, Franklin, Laser 128, Apple IIe or IIgs preferring to use slots 1 through 7 can now enjoy the speed and performance that they've been waiting for. With RamFactor, you'll be able to instantly add another 256K, 512K, or a full 1 MEG on the main board and an expansion connector can address an additional 16 MEG of memory for possible future expansion. Since virtually all software is automatically compatible with RamFactor, you'll immediately be able to load programs into RamFactor for instantaneous access to information. You'll also be able to store more data for larger word processing documents, bigger data bases, and expanded spreadsheets. And with the battery back-up option, you can switch on your Apple and your favorite software will be up and running in less than 1 second!



## Very Compatible

All the leading software is already compatible with RamFactor. Programs like AppleWorks, Pinpoint, BPI, Managing Your Money, Dollars and Sense, SuperCalc 3A, PFS, MouseWrite, MouseDesk, MouseCalc, Sensible Speller, Applewriter IIe, Business Works, ReportWorks, Catalyst 3.0 and more. And RamFactor is fully ProDOS, DOS 3.3, Pascal 1.3 and CP/M compatible. In fact, no other slot 1-7 memory card is more compatible with commercial software.

## AppleWorks Power

There are other slot 1-7 cards that give AppleWorks a larger desktop, but that's the end of their story. But RamFactor is the only slot 1-7 card that increases AppleWorks internal memory limits, increasing the maximum number of records in the database and lines permitted in the word processor, and RamFactor is the only standard slot card that will automatically load all of AppleWorks into RAM dramatically increasing speed and eliminating the time required to access the program disk, it will even display the time and date on the AppleWorks screen with any ProDOS clock. RamFactor will automatically segment large files so they can be saved on 5 1/4", 3 1/2", and hard disks. All this performance is available on the Apple II+,

IIe, Franklin, or Laser 128 when used with an 80 column display. RamFactor, no other standard slot card comes close to enhancing AppleWorks so much.

## The "Instant On" Apple.



With the RamCharger battery backup option, RamFactor will retain programs and data during power failures or even when you turn your computer off. And because RamFactor plugs into slots 1-7, it's the only memory card that can cold boot an Apple without using slow floppy or hard disks. An available "Y" cable allows one RamCharger to power two RamFactors.

## Powerful Program Switcher

With RamFactor, you can organize memory into multiple work areas and switch between them. Each work area can contain different programs and even different operating systems. Now you can switch from one program to another or even switch from AppleWorks to DOS 3.3 to CP/M to Pascal to ProDOS in under a second. And with our battery back-up option, you can have permanent storage for up to 20 years.

## Features:

- 256K to 1 MEG on main board. Expansion connector for possible further expansion to 16 MEG.
- Fully Apple II Memory Expansion compatible
- Compatible with Apple IIe, II+, IIgs, Franklin and Laser 128
- Battery back-up option allows you to turn on your Apple and run your favorite programs in less than 1 second!
- Automatically recognized by ProDOS, DOS 3.3, Pascal, and standard CP/M
- Built-in RamDrive™ software (a true RAM disk not disk caching)
- Systems are directly bootable from RamFactor if desired
- Built-in self diagnostic software
- Automatic expansion with AppleWorks 1.3 or 2.0
- Allows Apple II+ and IIe to run your AppleWorks without buying any additional software
- Accelerates AppleWorks
- Displays time and date on the AppleWorks screen with any ProDOS clock
- Fits any I/O slot except slot 3
- Fully socketed and user upgradeable
- Much, much more

RamFactor with 256K	\$239
RamFactor with 512K	\$279
RamFactor with 1 MEG	\$359
RamFactor with 2-16 MEG	CALL
RamCharger backup option	\$179
"Y" cable	\$24
(Allows one RamCharger to power two RamFactors.)	

Order RamFactor today . . . with 15 day money back guarantee and our "no hassle" five year warranty. See your dealer or call (214) 241-6060, 9 a.m. to 11 p.m., 7 days, or send check or money order to Applied Engineering, MasterCard, Visa and C.O.D. welcome. Texas residents add 6% sales tax. Add \$10.00 if outside USA.

**AE Applied Engineering™**  
The Apple enhancement experts.

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

## How to Clear the Back-Up Bit.....Bob Sander-Cederlof

I received a letter from Paul R. Santa-Maria today, with a very good question: "How is the backup bit in the file access byte cleared in ProDOS 8?" Paul is writing a program that can use the backup bit, but he needs to be able to clear it.

The information about this bit in the various reference manuals is contradictory and incomplete. Apple's ProDOS Technical Reference Manual (even the new ProDOS-8 edition) says:

ProDOS sets bit 5, the backup bit, of the access field to 1 whenever the file is changed (that is, after a CREATE, RENAME, CLOSE after WRITE, or SET\_FILE\_INFO operation). This bit should be reset to 0 whenever the file is duplicated by a backup program.

Note: Only ProDOS may change bits 2-4; only backup programs should clear bit 5, using SET\_FILE\_INFO.

As Paul pointed out in his letter, these two paragraphs contradict each other. Other references to "backup bit" listed in the index did not clear up the difficulty.

Paul noticed that one of the bytes in the System Global Page is called BUBIT (at \$BF95). The only explanation of this bit is that it can be changed before MLI calls, and a comment "BACKUP BIT DISABLE, SETFILEINFO ONLY".

Neither of us could find any further information in Apple's manuals, or even in the various third-party books.

I did get some help from the supplement to "Beneath Apple ProDOS", and also from my Apple itself. First I did a search of the ProDOS code while it was in RAM and found two references to \$BF95, at \$DE7A and at SF7EF. (These are the addresses in Version 1.1.1, and are slightly different from the addresses in Version 1.2, 1.3, and 1.4.) The first reference is at the general exit from all MLI calls, and it stores a zero at \$BF95 (BUBIT). The second is inside the SET FILE INFO processor. Here is a piece of the code:

```
F7EF- LDA BUBIT
      EOR #$20
      AND $FE7D      CURRENT ACCESS BITS
      AND #$20      ISOLATE BACKUP BIT
      STA $FEB4
```

## DON LANCASTER STUFF

### INTRODUCTION TO POSTSCRIPT

A 65 min user group VHS video with Don Lancaster sharing many of his laser publishing and Postscript programming secrets.

Includes curve tracing, \$5 toner refilling, the full Kroy Kolor details, page layouts, plus bunches more.

\$39.50

FREE VOICE HELPLINE

### ASK THE GURU

An entire set of reprints to Don Lancaster's ASK THE GURU columns, all the way back to column one. Edited and updated.

Both Apple and desktop publishing resources are included that are not to be found elsewhere.

\$24.50

### APPLE IIc/IIe ABSOLUTE RESET

Now gain absolute control over your Apple! You stop any program at any time.

Eliminates all dropouts on your HIRES screen dumps. Gets rid of all hole blasting. For any IIc or IIe.

\$19.50

### POSTSCRIPT SHOW & TELL

Unique graphics and text routines the others don't even dream of. For most any Postscript printer.

Fully open, unlocked, and easily adaptable to your own needs. Available for Apple, PC, Mac, ST, many others.

\$39.50

VISA/MC

Box 809-SC

SYNERGETICS

Thatcher, AZ 85552

(602) 428-4073

According to the BAP Supplement, \$FEB4 is later ORed into the Access Bits, immediately before the update is complete.

Apparently the steps necessary to clear the backup bit are:

- 1) read the current file information using GET FILE INFO;
- 2) clear the backup bit in the access byte and set at least bit 5 of \$BF95 to 1;
- 3) and use SET FILE INFO to install the change.

I wrote a test program to perform those steps, and it worked!

My program displays some information, so that I can see what it has done. Line 1170 reads the current file info and displays it in hex. The first byte displayed is the byte with the access bits. Lines 1180-1200 clear bit 5, the backup bit, in the access byte. Line 1210 changes BUBIT (\$BF95) from \$00 to \$FF, so that SET FILE INFO will not set the backup bit. Lines 1220-1240 call MLI to SET FILE INFO. Finally, lines 1260-1380 read the file info and display it again, to see if it worked.

To make my test program simple, I assembled the pathname of a file I knew was on the mounted volume. The pathname is in line 1480. You should substitute here the name of the file you really want to play with.

By the way, there is another way to clear the backup bit. You can read and write directory sectors directly, using the READ\_BLOCK and WRITE\_BLOCK calls. If you are writing a super snazzy backup program, you may want to do it this way. It can be easier to follow the directory tree using such direct access.

```
1000 *SAVE CLEAR.BUBIT
1010 *
-----  
BF00- 1020 MLI .EQ $BF00
BF95- 1030 BUBIT .EQ $BF95
1040 *
-----  
FBDD- 1050 BELL .EQ $FBDD
FD8E- 1060 CROUT .EQ $FD8E
FDDA- 1070 PRBYTE .EQ $FDDA
FDED- 1080 COUT .EQ $FDED
1090 *
-----  
1100 .MA MLI
1110 JSR MLI
1120 .DA #J1,J2
1130 BCS ERROR
1140 .EM
1150 *
-----  
1160 CLEAR.BUBIT
0800- 20 1B 08 1170 JSR GET.FILE.INFO.AND.DISPLAY.IT
0803- AD 46 08 1180 LDA INFO+3
0806- 29 DF 1190 AND #$DF CLEAR BACKUP BIT
0808- 8D 46 08 1200 STA INFO+3
080B- CE 95 BF 1210 DEC BUBIT BUBIT = $FF
080E- A9 07 1220 LDA #$07
0810- 8D 43 08 1230 STA INFO
0813- 1240 >MLI $C3,INFO SET INFO, CLEARING BUBIT
0813- 20 00 BF 0000> JSR MLI
0816- C3 43 08 0000> .DA #$C3,INFO
0819- B0 22 0000> BCS ERROR
```

```

1250 *-----  

1260 GET.FILE.INFO.AND.DISPLAY.IT  

081B- A9 0A 1270 LDA #$0A  

081D- 8D 43 08 1280 STA INFO  

0820- 20 00 BF 1290 MLI $C4,INFO READ AND DISPLAY NEW INFO  

0823- C4 43 08 0000> JSR MLI  

0826- B0 15 0000> .DA #$C4,INFO  

0828- A0 03 1300 BCS ERROR  

082A- B9 43 08 1310 .1 LDY #3  

082D- 20 DA FD 1320 LDA INFO,Y  

0830- A9 AE 1330 JSR PRBYTE  

0832- 20 ED FD 1340 LDA #".  

0835- C8 1350 JSR COUT  

0836- C0 12 1360 INY  

0838- 90 F0 1370 CPY #18  

083A- 4C 8E FD 1380 BCC .1  

083A- 4C 8E FD 1380 JMP CROUT  

1390 *-----  

083D- 20 DA FD 1400 ERROR JSR PRBYTE  

0840- 4C DD FB 1410 JMP BELL  

1420 *-----  

0843- 0A 1430 INFO .HS OA  

0844- 55 08 1440 .DA PATH  

0846- 1450 .BS 15  

1460 *-----  

0855- 06 1470 PATH .DA #LEN  

0856- 50 52 4F 1480 :AS /PRODOS/  

06- 44 4F 53 1490 LEN :EQ #-PATH-1  

1500 *-----  


```

0000 ERRORS IN ASSEMBLY

## **EnterSoft:**

Basic-like macros which make the complex simple. Don't re-write that multiplication routine for the hundredth time! Get EnterSoft instead! Do 8/16/32/64 bit Math/Input-Output/Graphics simply without all of the hassles. These routines are a must for the serious programmer who doesn't want to spend all of his/her time trying to re-invent the wheel. DOS 3.3 Version=\$30.00, ProDos Version=\$30.00, BOTH for only \$50.00. GET YOURS TODAY.

## **A Shape Table Program:**

For once! A shape table program which is logically organized into its component parts. Each section resides in its own program. The editor, disk access, Hi-Res section; each section is separate. Written almost entirely in Basic, it is easily modified. Not copyprotected! Put them on a Hard Disk, Ram Drive, anywhere! DOS 3.3 Version=\$20.00, ProDos Version=\$20.00, BOTH for \$30.00!



## Sean Nolan's Proposed Setup File Standard

Sean Nolan has written a very significant program, and released it to the public domain in the November 1987 issue of Call APPLE. SETUP.SYSTEM is a ProDOS-8 program to give startup features similar to those available in ProDOS-16.

The normal ProDOS-8 boot sequence simply installs the ProDOS image and then loads and runs the first type SYS file found in the main directory whose name ends with ".SYSTEM". Apple puts BASIC.SYSTEM (Applesoft) there, and I put the S-C Macro Assembler (SCASM.SYSTEM) there.

If you install SETUP.SYSTEM as the first SYStem file, it will gain control right after ProDOS is loaded. SETUP.SYSTEM scans the directory for a subdirectory named SETUPS. If one is found, all of the type SYS and type BIN files in that subdirectory will be executed, in the order found in the subdirectory. After all of those files have been executed, SETUP.SYSTEM will go back to the main directory and start up the first SYStem file which follows SETUP.SYSTEM.

I got kind of excited about this program, and took several hours to type it in. I installed it in my RamFactor card, and put PRODRIVE in the SETUPS directory. I have a non-standard clock-calendar card, so I also put a program called FIX.CLOCK in SETUPS which patches ProDOS so it can deal with my clock. You could put all kinds of neat things in there. Sean suggests B11 Bashams Diversi-Cache and Diversi-Hack. The requirements are few:

- \* Programs must not touch RAM between \$BD00 and BEFF.
- \* BIN programs must end with an RTS.
- \* SYS programs must end with a QUIT call.
- \* Programs must not go past \$B8FF during loading.

I think you will find Sean's program very useful. He suggests developers start using it, without modification so as to maintain compatibility. Since he put it in the public domain, it has made it fairly easy. I can't list the program here, because it has already been published in Call APPLE. However, I did include the source code on the AAL Monthly Disk for October. I will also list the 512 bytes in hexadecimal here, so you can punch it in the hard way if you want to.

I have printed a checksum for every row of 16 bytes. Of course you don't type in the checksum. Instead, run the program below after typing in all of the bytes. If any of the row checksums is wrong, you know you made a mistake in that row. Sean set his version up with a checksum balancer at the end for the rest of the 512 bytes, so you can check for errors by using the second checksumming program below too. It should give a result of "00" if all the bytes are correct.

```

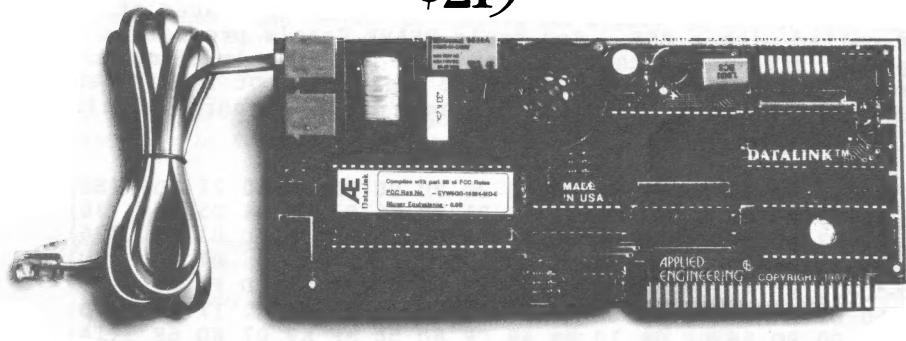
1000 *SAVE S.CHECKSUM
1010 *
1020 * Two CHECKSUM programs for testing your
1030 * entry of the SETUP.SYSTEM file. First
1040 * BLOAD SETUP.SYSTEM,A$2000, and then
1050 * execute either CHECKSUM program below.
1060 * If you use CS1, compare your results to
1070 * those printed in the newsletter. If you
1080 * use CS2, the single result should be 00.
1090 *-----*
00- 1100 PNTR .EQ $00,01
1110 *
FD8E- 1120 CROUT .EQ $FD8E
FDDA- 1130 PRBYTE .EQ $FDAA
FDED- 1140 COUT .EQ $FDED
1150 *
1160 * CHECKSUM 16 BYTES AT A TIME
1170 *
1180 CS1
0800- A9 00 1190 LDA #$2000 POINT TO FIRST PAGE TO BE SUMMED
0802- 85 00 1200 STA PNTR
0804- A9 20 1210 LDA /$2000
0806- 85 01 1220 STA PNTR+1
1230 *-----*
0808- 20 8E FD 1240 .1 JSR CROUT START A NEW LINE
080B- A5 01 1250 LDA PNTR+1 PRINT ADDRESS "XXXX-"
080D- 20 DA FD 1260 JSR PRBYTE
0810- A5 00 1270 LDA PNTR
0812- 20 DA FD 1280 JSR PRBYTE
0815- A9 AD 1290 LDA #"-"
0817- 20 ED FD 1300 JSR COUT
081A- A0 00 1310 LDY #0 FOR Y = 0 TO 15
081C- B1 00 1320 .2 LDA (PNTR),Y PRINT BYTE AS "XX "
081E- 20 DA FD 1330 JSR PRBYTE
0821- A9 A0 1340 LDA #"-"
0823- 20 ED FD 1350 JSR COUT
0826- C8 00 1360 INY NEXT Y
0827- C0 10 1370 CPY #16
0829- 90 F1 1380 BCC .2
1390 *-----*
082B- 20 ED FD 1400 JSR COUT PRINT " ("
082E- A9 A8 1410 LDA #"("
0830- 20 ED FD 1420 JSR COUT
0833- 88 00 1430 DEY Y=15
0834- A9 00 1440 LDA #0 FOR Y = 15 TO 0
0836- 51 00 1450 .3 EOR (PNTR),Y SUM = SUM.EOR.BYTE(Y)
0838- 88 00 1460 DEY NEXT Y
0839- 10 FB 1470 BPL .3
083B- 20 DA FD 1480 JSR PRBYTE PRINT "XX)"
083E- A9 A9 1490 LDA #")"
0840- 20 ED FD 1500 JSR COUT
1510 *-----*
0843- 18 00 1520 CLC BUMP POINTER TO NEXT GROUP
0844- A5 00 1530 LDA PNTR OF 16 BYTES
0846- 69 10 1540 ADC #16
0848- 85 00 1550 STA PNTR
084A- A5 01 1560 LDA PNTR+1
084C- 69 00 1570 ADC #0
084E- 85 01 1580 STA PNTR+1
0850- C9 22 1590 CMP /$2200 AT END YET?
0852- 90 B4 1600 BCC .1 ...NO
0854- 60 00 1610 RTS ...YES, FINISHED
1620 *
1630 * CHECKSUM ENTIRE 512 BYTES
1640 *
1650 CS2
0855- A2 00 1660 LDX #0
0857- 8A 00 1670 TXA
0858- 5D 00 20 1680 .1 EOR $2000,X checksum first page
085B- E8 1690 INX
085C- D0 FA 1700 BNE .1
085E- 5D 00 21 1710 .2 EOR $2100,X checksum second page
0861- E8 1720 INX
0862- D0 FA 1730 BNE .2
0864- 20 DA FD 1740 JSR PRBYTE print the result
0867- 60 00 1750 RTS
1760 *-----*

```

0000 ERRORS IN ASSEMBLY

# Global communications. Down-to-earth price.

## \$219



### Introducing the Datalink™ modem. Get instant access to networks and databases —plus network memberships and discounts worth \$177.95!

Now you can tap into a world of information — without draining your resources. Our Datalink™ modem lets your Apple IIGS, IIe or II+ communicate with other computers, download free software from networks and bulletin boards, access database services and more. For a remarkably low price.

#### The world at your fingertips.

With the Datalink modem, you'll be able to draw information from thousands of databases. Send and receive electronic mail — even overseas. Join clubs and bulletin boards to exchange software or solve computing problems with other Apple users. Download free public domain software. And share files with personal computers and mainframes.

#### There's nothing missing on Datalink.

Unlike the Smartmodem®? Datalink comes with its own easy-to-use communications software in ROM and on disk that supports macros, file transfers, on-line time display, data capture and datoscope mode. The software also allows you to store hundreds of phone numbers for auto dialing and log on.

Because Datalink is 100% Super Serial Card compatible, you can also run virtually all other communication software, including Point-to-Point, Apple Access and ASCII Express, to name a few.

The compact design allows Datalink to fit in any slot — including slots 1 and 2 of the IIGS with a cooling fan installed. Datalink operates at 1200 or 300 baud. Built-in diagnostics check for accuracy of data transmission, and Datalink lets you track the progress of calls either electronically or via an on-board speaker.

#### Introductory offers from popular networks included.

When you purchase Datalink, you'll get a fee-waived membership to The Source worth \$49.95, \$60.00 worth of free on-line time from News Net, A free \$50.00 subscription to the Official Airline Guide, A free subscription to the GENIE network worth \$18.00! And information on MCI Mail.

#### Order today!

To order or for more information, see your dealer or call Applied Engineering today, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 6 1/4% sales tax. Add \$10.00 outside U.S.A.

	Applied Engineering Datalink	Hayes™ Smartmodem 1200A
Price	\$219	\$439
Max. transmission rate	1200 baud	1200 baud
Warranty period	5 years at no charge	2 years or 4 years for \$75
Software included	YES	NO
Hayes AT command set	YES	YES
Help screens	YES	NO
On-board telephone jacks	YES	NO
Fits any slot (even with fan)	YES	NO

Smartmodem is a registered trademark of D.C. Hayes Corporation

**AE** Applied Engineering  
*The Apple enhancement experts.*

Box 798, Carrollton, TX 75006  
(214) 241-6060

Since writing this article, I have discovered that ProDOS-8 (versions 1.3 and later) has a new "secret" feature: before loading and executing the first type "SYS" file named "anything.SYSTEM" it searches the main directory for a type \$E2 file named "ATINIT". If ATINIT is there it is loaded at \$2000 and called with a JSR. You could BSAVE Sean's program as a type \$E2 file named "ATINIT" and save the trouble of making it first in the directory. I haven't tried it, but I think that would work without making any changes to his program at all.

2000-A2	01	8E	6F	BF	CA	BD	00	20	9D	00	BD	BD	00	21	9D	(8B)
?010-00	BE	E8	D0	F1	CA	9A	4C	21	BD	06	D3	C5	D4	D5	D0	(36)
2020-D3	AD	30	BF	8D	BF	BE	20	00	BF	C5	BE	BE	AD	01	BD	(36)
2030-29	0F	AA	E8	8E	00	BD	A9	2F	8D	01	BD	AD	04	BF	8D	(7B)
2040-D2	BD	AD	05	BF	8D	D7	BD	A2	02	BD	CB	BD	9D	F2	03	(98)
2050-BD	CA	BD	9D	03	BF	CA	10	F1	9A	20	85	BE	A2	17	A9	(5D)
2060-00	9D	58	BF	CA	10	FA	A9	CF	8D	58	BF	A9	07	8D	6F	(1A)
2070-BF	AD	82	C0	8D	OC	C0	8D	0E	C0	8D	00	C0	20	84	FE	(45)
2080-20	2F	FB	20	93	FE	20	89	FE	20	58	FC	20	00	BF	C6	(33)
2090-C5	BE	90	28	A2	0D	BD	F1	BE	9D	AC	05	CA	D0	F7	BD	(FA)
20A0-01	BD	09	80	9D	BB	05	E8	EC	00	BD	90	F2	A9	23	85	(C2)
20B0-24	A9	OB	20	24	FC	20	0C	FD	4C	89	BD	20	15	BE	B0	(EC)
?0C0-0D	20	00	BF	C6	C2	BE	20	8E	BE	4C	48	BD	18	EE	F4	(83)
?0D0-03	A9	00	8D	04	BF	A9	00	8D	05	BF	A9	00	8D	3E	BE	(A6)
20E0-8D	82	BE	A9	00	8D	D2	BE	20	15	BE	B0	1B	AE	80	02	(F5)
20F0-A0	06	BD	80	02	D9	0E	BE	D0	EE	CA	88	10	F4	EE	02	(84)
?100-BE	A9	FF	F0	E3	20	8E	BE	20	00	BF	65	C8	BE	2E	53	(1A)
?110-59	53	54	45	4D	20	00	BF	C8	D1	BE	B0	68	AD	D6	BE	(73)
?120-8D	D8	BE	8D	DD	BE	20	00	BF	CA	DC	BE	B0	57	AD	A3	(DB)
?130-02	8D	54	BE	AD	A4	02	8D	44	BE	EE	3E	BE	A9	00	A2	(7C)
?140-FE	E8	E8	C9	0D	90	05	ED	44	BE	B0	F5	A8	A9	04	88	(70)
?150-30	08	18	69	27	90	F8	E8	D0	F5	8D	D9	BE	8E	DA	BE	(CB)
?160-20	00	BF	CE	D7	BE	B0	1D	20	00	BF	CA	DC	BE	B0	15	(07)
?170-AD	80	02	F0	C5	29	0F	8D	80	02	AD	90	02	49	FF	F0	(4A)
?180-04	49	F9	D0	B5	08	20	00	BF	CC	CF	BE	28	60	AE	90	(8D)
?190-02	A9	20	E8	F0	06	AE	9F	02	AD	A0	02	8E	EC	BE	8D	(F8)
?1A0-ED	BE	20	00	BF	C8	E4	BE	B0	DB	AD	E9	BE	8D	EB	BE	(17)
?1B0-20	00	BF	CA	EA	BE	20	85	BE	B0	D2	6C	EC	BE	02	00	(44)
?1C0-01	BD	01	1A	BD	01	00	BD	04	00	00	00	00	00	00	01	(A3)
?1D0-00	03	1A	BD	00	B9	00	02	00	00	00	00	04	00	80	02	(99)
?1E0-27	00	00	00	03	80	02	00	B9	00	04	00	00	00	00	B1	(AA)
?1F0-00	00	C9	CE	D3	C5	D2	D4	A0	D6	CF	CC	D5	CD	C5	E8	(57)

Apple Assembly Line (ISSN 0889-4302) is published monthly by S-C SOFTWARE CORPORATION, P.O. Box 280300, Dallas, Texas 75228. Phone (214) 324-2050. Subscription rate is \$18 per year in the USA, sent Bulk Mail; add \$3 for First Class postage in USA, Canada, and Mexico; add \$14 postage for other countries. Back issues are \$1.80 each (other countries inquire for postage). A subscription to the newsletter and the Monthly Disk containing all source code is \$64 per year in the US, Canada and Mexico, and \$87 to other countries.

All material herein is copyrighted by S-C SOFTWARE CORPORATION, all rights reserved. (Apple is a registered trademark of Apple Computer, Inc.)